Downloaded from orbit.dtu.dk on: Jan 01, 2018

Technical University of Denmark

Large Scale Computing for the Modelling of Whole Brain Connectivity

Albers, Kristoffer Jon; Schmidt, Mikkel Nørgaard; Mørup, Morten

Publication date:
2017

Document Version
Publisher's PDF, also known as Version of record

Link back to DTU Orbit

Citation (APA):
Albers, K. J., Schmidt, M. N., & Mørup, M. (2017). Large Scale Computing for the Modelling of Whole Brain Connectivity. DTU Compute. (DTU Compute PHD-2017, Vol. 450).

DTU Library
Technical Information Center of Denmark

General rights
Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

Large Scale Computing for the Modelling of Whole Brain Connectivity

Kristoffer Jon Albers

Kongens Lyngby 2017
PhD-2017-450

Technical University of Denmark
Department of Applied Mathematics and Computer Science
Richard Petersens Plads, building 324,
2800 Kongens Lyngby, Denmark
Phone +45 4525 3031
compute@compute.dtu.dk
www.compute.dtu.dk

ISSN 0909-3192

Summary (English)

The human brain constitutes an impressive network formed by the structural and functional connectivity patterns between billions of neurons. Modern functional and diffusion magnetic resonance imaging (fMRI and dMRI) provides unprecedented opportunities for exploring the functional and structural organization of the brain in continuously increasing resolution. From these images, networks of structural and functional connectivity can be constructed. Bayesian stochastic block modelling provides a prominent data-driven approach for uncovering the latent organization, by clustering the networks into groups of nodes with a shared connectivity pattern.

Modelling the brain in great detail on a whole-brain scale is essential to fully understand the underlying organization of the brain and reveal the relations between structure and function, that allows sophisticated cognitive behaviour to emerge from ensembles of neurons. Relying on Markov Chain Monte Carlo (MCMC) simulations as the workhorse in Bayesian inference however poses significant computational challenges, especially when modelling networks at the scale and complexity supported by high-resolution whole-brain MRI.

In this thesis, we present how to overcome these computational limitations and apply Bayesian stochastic block models for un-supervised data-driven clustering of whole-brain connectivity in full image resolution.
We implement high-performance software that allows us to efficiently apply stochastic blockmodelling with MCMC sampling on large complex networks. To obtain the necessary computational performance, we find that both hardware and model specific properties must be taken into consideration - to an extend not supported by generic modelling tools. Computational overhead is reduced by an

approach, where key values are cached to avoid re-computations, while table-lookups are utilized for frequently computed special functions. The efficient memory-management of C++ is utilized to implement dedicated data-structures, optimized to facilitate performance-critical operations related to the inference procedure. Furthermore, the software is based on a modular design, which allows us to couple and explore different models and sampling procedures in runtime, still being applied to full-sized data.

Using the implemented tools, we demonstrate that the models successfully can be applied for clustering whole-brain connectivity networks. Without being informed of spatial information, the data-driven models can discover spatial homogeneous regions that are meaningful and in agreement with existing anatomical atlases.
We further demonstrate that structural and functional connectivity share information, allowing us to jointly model both modalities. For limited, noisy fMRI data we find that integrating structural information aids in discovering the functional organization better than using the fMRI data alone.

Though structure and function describes very different properties of the brain, we find that probabilistic modelling provides an intuitive data-driven approach for uncovering the latent organization in connectivity networks. We find that the stochastic block models can be computationally scaled to model whole-brain connectivity, and by doing so allows us to better utilize the full potential of high-resolution MRI and advances our understanding of both the functional and structural organization of the entire brain.

Summary (Danish)

Menneskets hjerne udgør et imponerende netværk, opbygget af strukturelle og funktionelle mønstre mellem milliarder af neuroner. Hjernens funktion og struktur kan i stadig højere opløsning kortlægges ved hjælp af magnetisk resonansbilleddannelse (funktionel og diffusions MRI), hvorfra netværk over hjernens strukturelle og funktionelle forbindelser kan genereres. Bayesiansk stokastisk blokmodellering repræsenterer en databaseret tilgang til at afdække den skjulte organisering, ved at opdele netværk i grupper, der udelukkende er baseret på delte forbindelsesmønstre.

Modellering af hele hjernen i høj opløsning er afgørende for fuldt ud at kortlægge den underliggende opbygning af hjernen og fastslå forholdet mellem struktur og funktion, som tillader sofistikert kognitiv adfærd at udspringe fra grupper af neuroner. Metoder baseret på Markovkæde Monte Carlo simulering (MCMC) udgør en grundsten i bayesiansk inferens, men lider under at være meget beregningskrævende - især til at modellere netværk i den størrelse og kompleksitet, som opnås for en høj opløsning over hele hjernen.

I denne afhandling præsenterer vi hvordan man kan overvinde disse beregningskrævende begrænsninger og anvende bayesianske stokastiske blokmodeller til at beskrive hele hjernen i den høje opløsning, som understøttes af moderne MRI billeddannelse.

Vi implementerer højt-ydende software, der giver os mulighed for at benytte stokastisk blockmodellering med MCMC til effektivt at modellere meget store, komplekse netværk. Vi opnår en høj ydelse ved at optimere implementeringen i forhold til både specifikke egenskaber i hardware og model specifikation. Vi

udnytter især hukommelses-optimering og benytter tabelopslag til at minimere beregningsomkostningerne ved ofte brugte specialfunktioner. Når programmet derudover er opbygget modulært, opnår vi et design, der tillader os at afprøve forskellige kombinationer af modeller og inferensprocedurer på data i fuld skala.

Vi demonstrerer ved hjælp af de implementerede værktøjer, at de afprøvede modeller kan anvendes til at finde gruperinger i konnektivitets netværk over hele hjernen. De datadrevne modeller kan uden at være oplyst om rumlige informationer, opdage rumlige homogene regioner, der er troværdige og stemmer overens med eksisterende anatomiske atlas. Vi demonstrerer endvidere, at der er delt information i strukturel og funktionel konnektivitet, hvilket giver os mulighed for at modellere begge modaliteter samtidigt. Vi viser, at for begrænset og støjfyldt fMRI data er integration af strukturel information medvirkende til, at opdage den funktionelle organisering bedre end udelukkende at anvende fMRI data.

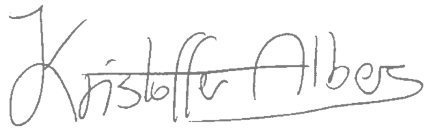
Selvom struktur og funktion beskriver meget forskellige egenskaber i hjernen, kan vi se, at probabilistisk modellering er en intuitiv databaseret tilgang til at afdække hjernens latente underliggende organisering. Vi konkluderer, at de stokastiske blokmodeller beregningsmæssigt kan skaleres til at modellere forbindelser i hele hjernen. Dette giver muligheder for bedre at udnytte det fulde potentiale af højt opløste MRI billeder og udvide vores forståelse af hjernens funktionelle og strukturelle opbygning.

Preface

This thesis was prepared at the Section for Cognitive Systems at the Department of Applied Mathematics and Computer Science, Technical University of Denmark in partial fulfilment of the requirements for acquiring the Ph.D. degree in engineering. The project was jointly funded by the Lundbeck foundation (two thirds) through the Brain connectivity project and by DTU Compute. My main supervisor was Associate Professor Mikkel N. Schmidt, DTU Compute. My co-supervisor was Associate Professor Morten Mørup, DTU Compute.

The thesis deals with designing and implementing software that overcomes the computational challenges for modelling whole-brain connectivity in large-scale. Aside from implementing software, the thesis consists of three published conference papers, one submitted journal paper, and two papers in preparation. The studies were conducted between November 2013 and April 2017.

Kgs. Lyngby, 02-04-2017



Kristoffer Jon Albers

Acknowledgements

First of all, I want to thank my supervisors, Mikkel N. Schmidt and Morten Mørup. Thank you for giving me the unique opportunity to be part of the Brainconnectivity project, and thank you for your patience, support and guidance in all matters.

I want to thank all members of the Brainconnectivity project and everyone I have worked and published with. I also thank the Lundbeck foundation for supporting and funding the Brainconnectivity project that my Ph.D. is part of.

I want to thank the entire section for cognitive systems. Despite the diversity of research projects, the entire section is embodied by a fantastic team spirit, where everyone fits in to create an inclusive and vibrant yet professional atmosphere.

Finally, I want my family and friends to know that the challenges of writing this thesis would have been much tougher if it wasn't for your continuous encouragement, cheering and support.

List of Publications

Manuscripts included in this thesis

A *The Influence of Hyper-parameters in the Infinite Relational Model.* Kristoffer J. Albers, Morten Mørup, and Mikkel N. Schmidt. Published in *Machine Learning for Signal Processing, IEEE International Workshop on, (MLSP), 2016.*

B *Numerical Approximations for Speeding Up MCMC Inference in the Infinite Relational Model.* Mikkel N. Schmidt and Kristoffer Jon Albers. Published in *European Signal Processing Conference (EUSIPCO), 2015.*

C *Local Modes in the Posterior Distribution of Dirichlet Process Mixture Models.* Kristoffer Jon Albers, Morten Mørup, and Mikkel N. Schmidt. (preliminary work).

D *Nonparametric Bayesian Clustering of Structural Whole Brain Connectivity in Full Resolution.* Karen Sandø Ambrosen, Kristoffer Jon Albers, Tim B. Dyrby, Mikkel N. Schmidt, and Morten Mørup. Published in *Pattern Recognition in NeuroImaging (PRNI), 2014.*

E *Predictive Validation of Human Brain Parcellation.* Karen Sandø Ambrosen, Kristoffer Jon Albers, Matthew G. Liptrot, Tim B. Dyrby, Mikkel N. Schmidt, and Morten Mørup. (Submitted, under review).

F *Functional Whole-Brain Parcellation Improved by the Inclusion of Structural Connectivity* (preliminary work)

Not included in this thesis

The following journal paper has been prepared as part of the project, but is not included in this thesis. It is however sparingly referred to and illustrates that the methodology and implemented tools intuitively can be utilized in different research areas. As it is not yet published it is included as supplementary information in Appendix H.

H.1 *Predictive Evaluation of Human Value Segmentations.* Kristoffer Jon Albers, Morten Mørup, Mikkel N. Schmidt, and Fumiko Kano Glückstad. (Submitted, under review).

Nomenclature

Abbreviations and fixed symbols for stochastic blockmodelling

SBM	Stochastic Block Model
IRM	Infinite Relational Model
CRP(α)	Chinese Restaurant Process, with concentration parameter α.
$B(a, b)$	Beta function with parameters a and b.
$\Gamma(a)$	Gamma function with parameter a.
K	Total number of clusters.
J	Total number of nodes.
$\boldsymbol{A}$	Adjacency matrix.
$\boldsymbol{A}^{(s)}$	Adjacency matrix for subject s.
$\boldsymbol{z}$	Clustering (cluster assignment vector).
z_i	Cluster assignment for node i.
$\boldsymbol{z}_{\setminus i}$	Clustering, ignoring assignment for node i.
$N^{+}_{\ell m}$	Link count between clusters ℓ and m.
$N^{-}_{\ell m}$	Count for possible but not observed links between ℓ and m.
$N^{+\setminus i}_{\ell m}$	Link count between ℓ and m, ignoring node i.
$N^{-\setminus i}_{\ell m}$	Non-link count between ℓ and m, ignoring node i.
n_ℓ	Number of nodes assigned to cluster ℓ.
$n_\ell^{\setminus i}$	Number of nodes assigned to cluster ℓ, ignoring node i.
$r^{+}_{i\ell}$	Number of links from node i to all nodes in cluster ℓ.
$r^{-}_{i\ell}$	Number of non-links from node i to all nodes in cluster ℓ.
$\widehat{r}^{+}_{i\ell}$	Number of links to node i from all nodes in cluster ℓ.
$\widehat{r}^{-}_{i\ell}$	Number of non-links to node i from all nodes in cluster ℓ.

Functions and data-structures

gammaln(a)	Computing the logarithm of the gamma function, $log(\Gamma(a))$
betaln(a,b)	Computing the logarithm of the beta function, $log(B(a,b))$.
N^+	Matrix of link counts for all pairs of clusters.
N^-	Matrix of non-link counts for all pairs of clusters.
$N^{+\setminus i}$	Matrix of link counts for all pairs of clusters, ignoring node i.
$N^{-\setminus i}$	Matrix of non-link counts for all pairs of clusters, ignoring node i.
r_i^+	Vector of link counts from node i to all clusters, $r_i^+ = [r_{i0}^+, ..., r_{iK-1}^+]$.
r_i^-	Vector of non-link counts from i to all clusters, $r_i^- = [r_{i0}^-, ..., r_{iK-1}^-]$.
$\widehat{r}_i^+$	Vector of link counts to node i from all clusters, $\widehat{r}_i^+ = [\widehat{r}_{i0}^+, ..., \widehat{r}_{iK-1}^+]$.
$\widehat{r}_i^-$	Vector of non-link counts to i from all clusters, $\widehat{r}_i^- = [\widehat{r}_{i0}^-, ..., \widehat{r}_{iK-1}^-]$.

Algorithmic time and memory complexities

$\Theta(f)$	Asymptotic bound of function f (both above and below).
$O(f)$	Asymptotic upper bound of the function f.

Contents

Chapter 1

Introduction

Network science plays a prominent role in multiple research areas, as many systems both naturally occurring and engineered can be described as complex networks. A core task in modelling networks is to cluster nodes with similar connectivity patterns, in order to uncover the latent connectivity structure of the network. Bayesian modelling provides a framework of statistical inference, with stochastic blockmodelling [Nowicki and Snijders, 2001] being a prominent approach for relational clustering of network data. Bayesian inference is a difficult and computationally time consuming problem [Cooper, 1990], often involving various variables and properties that cannot easily be analytically evaluated. Instead of relying on exact inference, the common approach is to utilize generic approximative approaches, with Markov Chain Monte Carlo (MCMC) algorithms providing an array of well-proven and successful sampling procedures [Bessiere et al., 2013]. Facilitated by the advance of raw computer power and data collection platforms, we can now obtain and study huge network containing millions or even billions of nodes, for which many of the properties previously studied on small networks might not apply [Newman, 2003]. This trend calls for scientists to explore the limitations of existing procedures and propose new methods when faced with large complex data sets [Lazar, 2013].

One of the most fascinating systems that can be described as a network is the human brain. Both in terms of its structural organization and functional capabilities it is one of the most complex objects contemplated in the universe.

It consists of tens of billions of neurons connected by trillions of transmission points [Baars and Gage, 2010], and is responsible for all aspects of human behaviour; from interpreting sensory inputs to controlling motor movements and cognitive processes. In recent years, modern techniques for brain imaging has provided *in vivo* and non-invasive means of studying the brain. Diffusion and functional magnetic resonance imaging (MRI) techniques has become prominent approaches for obtaining detailed images of the structural and functional organization of the brain - in steadily higher resolutions. Modelling brain connectivity in great detail and on the whole brain level is key to unveiling how the brain operates and allows sophisticated behaviour to arise from ensembles of neurons [Sporns et al., 2005]. This thesis is completed as part of the larger research project "Nonparametric Relational Modeling of Structural and Functional Brain Connectivity" (funded by the Lundbeck Foundation, 2012-2017)[1], which aims at inferring global structural and functional brain connectivity using Bayesian relational modelling. The thesis is motivated by the need for developing tools that computationally can handle the modelling of large complex networks, particular with focus on networks of brain connectivity.

Previous studies have shown that stochastic relational modelling with MCMC methods can be computationally implemented to handle very large networks. This is achieved by creating dedicated implementations, relying on hardware optimizations such as extensive parallelization using GPU [Hansen et al., 2011], or various algorithmic optimizations [Albers et al., 2013] as well as improved model techniques [Zhu et al., 2009] to speed up the computations. Designing and implementing such dedicated solutions can be a time-consuming process, where the programmer needs to optimize towards the specific hardware and memory architecture which is often not exposed in higher level programming and modelling languages. General and user-friendly software is a must to facilitate the use of Bayesian methods [Berger, 2000]. With the current trend in data complexity, we see a need for such tools to be both flexible and high-performance in order to allow various models and sampling strategies to be easily tested on large complex networks, such that the scientist will not have to spend significant time on algorithmic optimizations or rely on sub-sampled data, that might not reveal the same properties as full sized data.

The aim of this thesis is to facilitate the usage of Bayesian relational modelling as a data-driven approach for quantifying the functional and structural organization of the brain. In particular, it has been studied how to obtain the necessary computational performance for applying the data-driven stochastic block modelling framework to networks of whole-brain connectivity, at the size and complexity that can be obtained from MRI neuroimages in full image-resolution.

[1] `https://brainconnectivity.compute.dtu.dk/`

1.0.1 Software contribution

A main contribution of this thesis is the design and implementation of computational tools that can scale Bayesian relational modelling to handle such large networks of brain connectivity. The implemented tools are delivered as a stand-alone toolbox implemented in C++. The source code is maintained and documented at `https://github.com/kristofferalbers`. The design of the toolbox is based on an object oriented, modular programming scheme, that is defined to address the two major design aspects of the application. It allows high-performance MCMC inference on large complex networks, while easily allowing the user to set up different MCMC sampling procedures and modify the implemented models.

1.0.2 Included publications

The implemented tools are utilized in two different ways in the included work. Papers A, B and C explore various model and computational properties of stochastic blockmodelling in general, while Papers D, E and F utilizes the implemented tools as a practical approach for modelling whole-brain connectivity.
Paper D proves the concept of clustering whole-brain structural connectivity in full image resolution, using the Infinite Relational Model (IRM) [Kemp et al., 2006, Xu et al., 2006] which is a non-parametric extension to the stochastic blockmodel, capable of inferring an appropriate number of clusters from data.
Paper E presents a predictive framework for quantifying the quality of brain parcellations by statistical predictions on hold-out data, illustrating some of the advantages of Bayesian relational modelling.
Paper F utilizes this predictive framework for evaluating the quality of parcellations derived when integrating both whole-brain structural and functional information, compared to modelling a single modality.

1.0.3 Structure of the thesis

The thesis is structured as follows:

Chapter 2 presents how the brain can be represented as a complex network and introduces the problem of modelling whole brain connectivity.

Chapter 3 describes the theoretical framework of stochastic blockmodelling with MCMC inference and presents our sampling and model evaluation strategy.

Chapter 4 discusses elements of statistical software design and presents the design paradigm for the implemented toolbox.

Chapter 5 presents implementational details for realizing the toolbox design to allow the high-performance Bayesian modelling conducted in the included work.

Chapter 6 summarizes the research contributions of the project, presented in the included work.

Chapter 7 discusses and concludes on the project.

Chapter 2

Brain connectivity

This chapter first gives a short introduction to modern neuroscience as an interdisciplinary science, to see how neuroscience benefits from research in multiple areas beyond biology and medicine, such as mathematics, statistical machine learning and scientific computing. By looking at the anatomy and function of the brain, we see why it intuitively makes sense to represent the brain as a complex network and introduces the problem of modelling brain connectivity.

From looking at the MRI scanning procedure, we see how networks of brain connectivity are obtained which clarifies the complexity that must be faced when modelling whole brain connectivity based on high resolution images.

2.1 Neuroscience

Neuroscience is a very broad field of research, concerned with studying all aspects of the nervous system. Figure 2.1 presents three basic levels for studying the nervous systems of the brain. On the *cellular level*, applied molecular biology provides detailed descriptions of the physiological and morphological properties of the individual neuron. On the *systems level*, the complex interactions of neurons and regions are studied to understand the structure that allows for

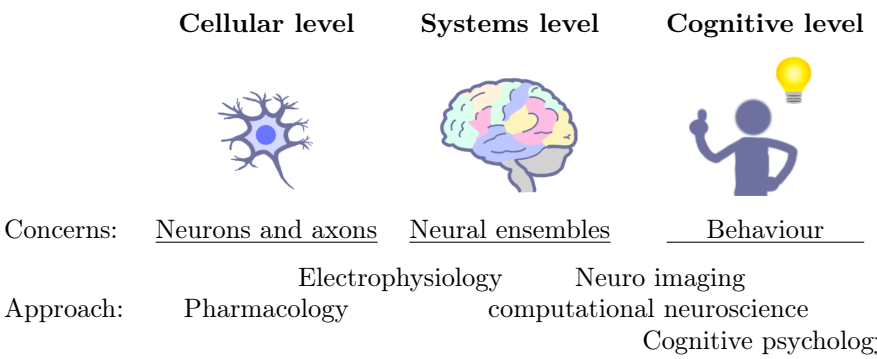


Figure 2.1: *General levels for studying the brain with examples of key experimental approaches.*

sophisticated functions such as specific sensory, memory and motor skills and forms the basis for behaviour and cognitive functions, studied at the *cognitive level*.

Though the biological structure and function of the individual neuron is well understood, descriptions of the way cognitive and behavioural functions arises from ensembles of neurons is yet highly speculative [Sporns et al., 2005]. The brain consists of a multitude of localized regions that can be studied individually. Sophisticated functions, however, rely on the collaboration of multiple regions, and some functionality only emerges as a result of the entire connectome [Yuste and Church, 2014].

Understanding the structural and functional properties in great detail and on the whole brain scale is essential and has received substantial focus in recent major research efforts.
In the *Brain Research through Advancing Innovative Neurotechnologies Initiative (BRAIN)* a priority research area is mapping the brain at various resolutions from single synapses to the whole brain [Jorgenson et al., 2015]. The *Human Connectome Project (HCP)* aims at mapping the entire connectome in high resolution and facilitates advances in image technique and processing [Glasser et al., 2016]. The *Human Brain Project (HBP)* has significant focus on research within information and communication technology in order to advance computational neuroscience and develop a related research infrastructure [Amunts et al., 2016], highlighting the importance for computational tools for both data processing and large scale modelling of structural and functional connectivity [Dayan and Abbott, 2005].

2.2 Network modelling of brain connectivity

An extensive map of brain connectivity is called a *connectome* [Hagmann, 2005, Sporns et al., 2005]. On a macroscopic scale the connectome can be conceived as a network, where vertices represent cortical or sub-cortical regions while edges represent pairwise relations between these regions, based on either the physical structure or functional activity [Daducci et al., 2012]. A common modelling goal in network science is to identify groups of vertices that share similar connectivity patterns. For a connectome this constitutes to cluster regions in the brain that appears to be organized structurally similar or behave functionally similar.

Figure 2.2 illustrates the pipeline for network modelling of brain connectivity, which can be separated into two distinct problems that will be discussed subsequently:

1) Obtaining data and constructing the connectivity networks. This procedure is not the focus in this thesis and will hence only be shortly outlined.

2) Obtaining and assessing clusterings of brain connectivity based on the connectivity networks.

2.2.1 Data acquisition and network construction

A network of *structural connectivity* is based on how the brain is physically linked. Depending on the resolution, the measured links can describe the connectivity formed by single synapses, fiber pathways or entire regions and centres in the brain. On shorter timescales (not influenced by neural plasticity), the structural connectivity of the brain appears to be static. Diffusion magnetic resonance imaging (dMRI) provides an *in vivo* and non-invasive approach for obtaining images of white matter tracts as a measure for the structural organization of the brain [Ghosh and Deriche, 2016]. dMRI is sensitive to the random (Brownian) motion of water, which allows it to indirectly map out the geometry of the brain tissue that restricts the free diffusion of water molecules. One pipeline of connectomics [Hagmann, 2005] combines dMRI and tractography with network science to model and study whole-brain structural connectivity [Sporns et al., 2005]. A network can here be constructed from dMRI, such that the nodes are based on segmented cortical areas while the links are based on white matter tracts constructed by tractography [Buchanan et al., 2012].

A network of *functional connectivity* can be constructed based on similarities in the temporal activity patterns of different regions (voxels) in the brain. In contrast to the structural organization, the neural activity in different areas in

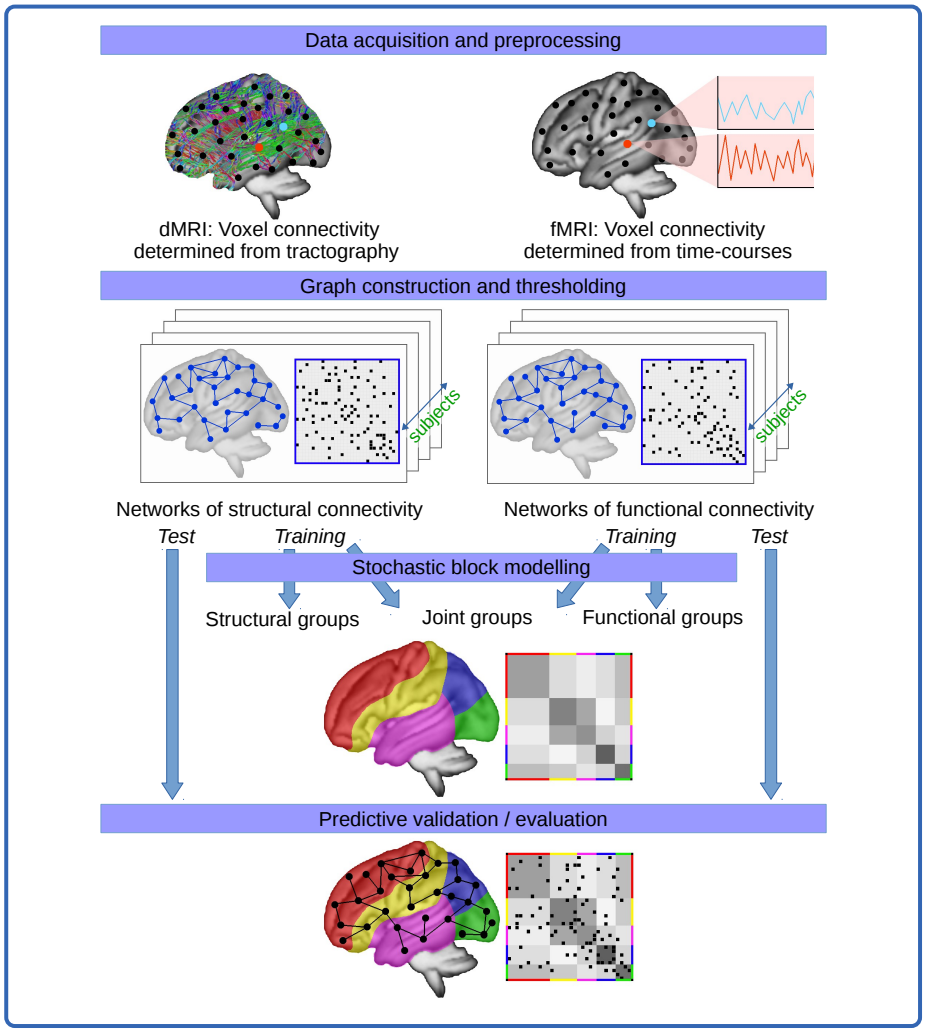


Figure 2.2: Pipeline for modelling of whole-brain connectivity. *Networks of structural and functional connectivity are obtained from dMRI and fMRI images. The networks can be split into training and hold-out test data. From training data parcellations of the brain can be obtained either from structure or function alone or from joint modelling. Based on their ability to statistically predict structure in the test data, the inferred parcellations can be valued and compared.*

the brain fluctuates which causes the functional connectivity patterns to continuously change [Sporns, 2009]. Neural activity causes increased energy usage for the active brain cells which correlates with an increased blood flow to the active regions in the brain. Monitoring changes in the blood flow can hence be used as an indirect measure of functional neural activity. Functional magnetic resonance imaging (fMRI) is sensitive to the difference in the magnetic properties of oxygenated and deoxygenated hemoglobin as an indirect measure of blood flow [Ogawa et al., 1990]. Continuously monitoring the blood oxygenation level dependent (BOLD) response, fMRI can be used to map out the temporal change in activity for different regions of the brain, which in turn can be used for constructing networks of functional connectivity. Such networks can for instance be constructed from the cross-correlation between the time series for measured neuronal activity at different regions [Bullmore and Sporns, 2009, Richiardi et al., 2013].

The constructed networks can be binarized by thressholding at a certain density and only keeping the strongest links. Even though function and structure describes completely different properties of the brain, the network construction allows both modalities to be represented by a similar specification. This allow us to use the same models to infer both structural and functional parcellations and evaluate the modalities in comparison.

2.2.2 Clustering of brain connectivity networks

For the modelling strategy, we use the stochastic framework presented in chapter 3, where the connectivity networks are modelled using stochastic blockmodelling (SBM) [Nowicki and Snijders, 2001]. The model parameters are inferred using Markov Chain Monte Carlo sampling, which is popularly applied for Bayesian inference [Zhu et al., 2009, Palla et al., 2012, Miller et al., 2009], and provides a data-driven approach for inferring clusterings of brain connectivity.
SBM takes both the link structure within and between clusters into account. Nodes are hence grouped when sharing similar connectivity patterns throughout the entire network, which is more explanatory than only identifying groups based on within cluster link densities [Fortunato and Barthelemy, 2007].

The model intuitively allows a single clustering to be inferred from multiple networks. This can either be modelled such that link densities between clusters are shared across networks [Andersen et al., 2012] or considered independent [Mørup et al., 2010]. In Paper E we use the first approach to model different sized populations based only on structural connectivity networks. In Paper F we use the second approach to jointly model networks of structural and functional connectivity.

In Paper D we use the Infinite Relational Model (IRM) which is a nonparametric extension of SBM that can infer an appropriate number of clusters from data [Kemp et al., 2006, Xu et al., 2006].
Chapter 4 and 5 presents the key concepts for the design paradigms and implementational details that allows for developing computational tools for generically applying the stochastic block models with a customisable MCMC sampling strategy, while still allowing modelling of whole-brain networks in high resolution.

The performance of the model is evaluated by splitting the data into training and test networks. SBM is a statistical generative model which provides salient evaluation of the predictive performance. The fit of the model can be evaluated by how well a clustering inferred from one network predicts other networks. Predictions can also be used for model selection. By comparing the predictive performance for clusterings inferred with different number of clusters, an appropriate number of clusters can be determined for SBM. To evaluate an inferred clustering, it can be compared with other clusterings that are either inferred from rescans of the same subject (Paper D), from other subjects or from atlases based on anatomical landmarks (Paper E).

2.3 Data

The data size poses major challenges for scaling SBM and IRM to model high-resolution whole-brain networks. In Paper D we use dMRI data presented in [Reislev et al., 2012] from which whole-brain structural connectivity graphs were obtained with 167.635 nodes and around one million links. In Papers E and F high-resolution data for multiple subjects from the Human Connectome Project (HCP) [Van Essen et al., 2013, Moeller et al., 2010, Feinberg et al., 2010, Setsompop et al., 2012, Xu et al., 2012] database was used. These networks contain 59,412 nodes of the cortex, excluding the medial wall.
Besides the data that is modelled in the included publications, connectivity networks are used in chapter 3 and 5 to asses some aspects of modelling and computational performance.
From the HCP data, we use structural connectivity networks for single subjects. These networks are used to compare SBM and IRM for different sampling strategies. Based on the structural connectivity graphs presented by Hagman et al. [Hagmann et al., 2008], a single network with 998 nodes is obtained by averaging the graphs for five different subjects. This network is binarized and symmetrized and is used to assess different sampling procedures and computational performance.

Chapter 3

Bayesian block modelling

The goal of the clustering problem is to partition the nodes of a graph into homogeneous clusters, based on the nodes structural similarities. Some clustering approaches are based on Bayesian statistics, including the stochastic blockmodel (SBM) [Snijders and Nowicki, 1997, Nowicki and Snijders, 2001] and its non-parametric extension, the Infinite Relational Model (IRM) [Kemp et al., 2006, Xu et al., 2006] that we have successfully used to partition large networks of whole brain connectivity.

This chapter first gives a short introduction to Bayesian modelling, Markov Chain Monte Carlo simulations and model evaluation in general before introducing the particular models and inference procedures we have utilized.

3.1 The Bayesian method

The Bayesian method can be dated back to the mathematical philosophy of reverend Thomas Bayes. By addressing the question of inferring causes from observed effects, Bayes envisioned a method to estimate the then called inverse probability; going from known frequencies of sampled data to the estimated probability of the underlying cause [Kadry, 2014].

The practical framework of Bayesian modelling was in large developed and popularised by the french mathematician Pierre-Simon Laplace towards the end of the 18th century. Motivated by his desire to describe celestial mechanics [Berger, 1997], Laplace often had to rely on combining ancient and modern astronomical observations of varying quality [Gillispie et al., 2000], illustrating the need to model the uncertainty caused by noise, differences and imprecisions in the data-measurement processes. This can be achieved by probabilistic (Bayesian) modelling, which to this day is widely used for providing a coherent way of representing and manipulating uncertainty within models [Ghahramani, 2013] and obtaining common-sense interpretations of the statistical conclusions [Gelman et al., 2014].

3.1.1 Bayesian modelling

Bayes' theorem describes a simple rule governing conditional probabilities:

$$p(\theta|X) = \frac{p(X,\theta)}{p(X)} = \frac{p(X|\theta)p(\theta)}{p(X)} \tag{3.1}$$

Bayesian modelling aims at learning unknown parameters of interest θ from a known set of observations X, utilizing Bayes theorem and the rules of probability:

$$posterior = \frac{likelihood \cdot prior}{evidence} \tag{3.2}$$

The likelihood function $p(X|\theta)$ describes the probability of observing the data given the parameters. The parameters are considered fixed, but are not observed directly. Our uncertainty about their values can be modelled by considering them as random variables. The prior function $p(\theta)$ hence describes our initial belief in the parameters before observing the data. Any parameters of the prior are called hyper-parameters. Our posterior belief is based on taking both our initial belief and the observed data into account. The full posterior distribution is hence the conditional distribution of the prior and likelihood:

$$p(\theta|X) = \frac{p(X|\theta)p(\theta)}{\sum_\theta p(X|\theta)p(\theta)} \tag{3.3}$$

where the evidence (marginal likelihood) is a normalization constant, summing over all possible values of θ; $p(X) = \sum_\theta p(X,\theta) = \sum_\theta p(X|\theta)p(\theta)$ for discrete θ and $p(X) = \int p(X|\theta)p(\theta)\, d\theta$ for continuous θ.

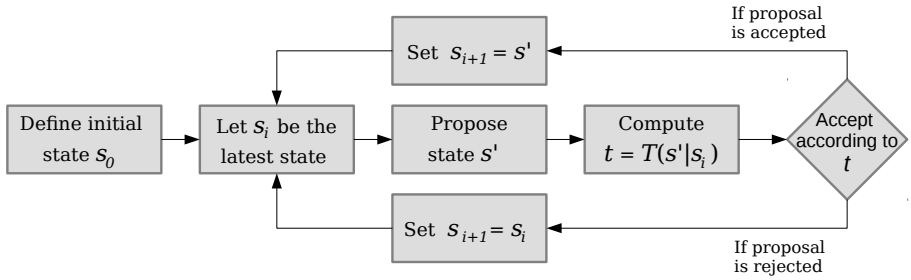


Figure 3.1: Procedure to generate a Markov Chain. *From an initial state s_0 the procedure iteratively proposes new states s' only depending on the current state s_i. The proposal is randomly accepted or rejected according to the transition probability $t = T(s'|s_i)$.*

In many situations it is unfeasible to evaluate the evidence analytically. This includes mixture models such as the blockmodels we use for clustering, where the evidence might yield exponentially many modes [Beal, 2003]. Instead the practical resolution is to resort to various approximation techniques, one being sampling-based Markov Chain Monte Carlo.

It is often computationally convenient to use a conjugate prior, being a prior distribution that combined with the likelihood yields a distribution of the same family as the posterior.

3.1.2 Markov Chain Monte Carlo

In many situations obtaining the exact posterior distribution is analytically or technically unmanageable. In such situations the model parameters can be inferred using Markov Chain Monte Carlo (MCMC) algorithms. By constructing the Markov Chain with the state space being the parameters of the model and the stationary (equilibrium) distribution being the target posterior, MCMC can simulate draws from the complex posterior distribution in the limit of a large enough number of samples.

Figure 3.1 shows the iterative procedure for computing a Markov Chain. From an initial state s_0 the procedure samples a sequence of states $\{s_0, s_1, \ldots, s_n\}$, such that each sampled state s_i only depends on the previous state s_{i-1}. For any pair of states s and s' the transition probability $T(s'|s)$ gives the probability for accepting a transition from s to s'. A state in the Markov Chain is nothing but an element of the model parameter space, being a possible configuration for

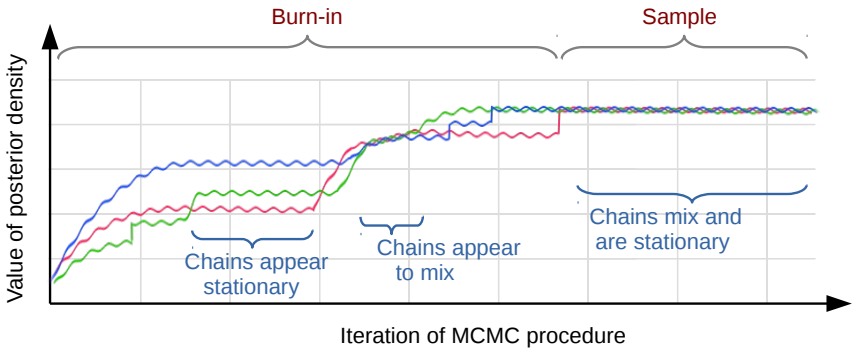


Figure 3.2: Convergence criteria. *In order to asses convergence multiple sequences of the MCMC procedure can be run in parallel. The sequences will not converge to the same distribution unless they appear both stationary and mix together. Ideally all samples before the sequences converges should be treated as burn-in.*

all model parameters. For a clustering model, the state describes a particular partition of the nodes as well as values for any other parameters and hyper-parameters that defines the model.

The early iterations of MCMC reflects the starting state rather than the target distribution and should be rejected as *burn-in* such that ideally all considered samples will be from the correct distribution. In advance it is however not possible to predict the length of a Markov Chain before it converges to the correct target distribution. Even though there exists methods for statistically assessing convergence [Mengersen et al., 1999, Cowles and Carlin, 1996] it is often easier to determine whether a chain has *not* converged. In practice lack of convergence is often determined by comparing multiple sequences run in parallel [Gelman and Rubin, 1992a, Gelman and Rubin, 1992b]. As illustrated in figure 3.2 the sequences must both individually appear stationary and mix together in order to suggest convergence [Gelman et al., 2014]. Stationarity and mixing can be determined by keeping track of the within-sequence and between-sequence variation of the sampled posterior values during the MCMC sampling.

Due to the computationally intensive and sequential nature of MCMC algorithms, they however dictate some practical and technical challenges that must be addressed - in particular when applied for large scale problems. Given enough samples the MCMC methods will give exact approximations though the needed number of samples might be unattainably large [Beal, 2003]. For large problems such as clustering of large complex networks it might not be possible to draw enough samples to explore the full posterior distribution or even finish the com-

plete burn-in sequence. In Paper C we have explored the posterior distribution of Dirichlet process mixture models in order to visualize and count the local modes that might trap the MCMC sampling procedure and examined the practical usefulness of the inferred clusterings when convergence cannot be reached or the MCMC procedure get stuck in local modes. In chapter 5 we discuss some of the technical and algorithmic details for designing and implementing large scale MCMC software, taking modern computer architecture into account.

3.1.3 Model evaluation

The performance of a given model can be evaluated by its ability to recover or predict data that was not included in the modelling, as a measure for how well the model can recover structure from data.

3.1.3.1 Mutual Information

If ground truth is available for a clustering problem, the inferred clustering can be compared with the true clustering. We use the permutation invariant measure of mutual information (MI) to compare clusterings. Mutual information is an informational theoretic measure for the interdependency between two variables. The mutual information for two discrete random variables X and Y is computed as:

$$MI(X,Y) = \sum_{x \in X} \sum_{y \in Y} p(x,y) \log \left(\frac{p(x,y)}{p(x)p(y)} \right),$$

where $p(x)$ and $p(y)$ are the marginal probabilities of $X = x$ and $Y = x$ respectively, while $p(x,y)$ represents the joint probability distribution.

When computing the mutual information between the true clustering Z and an estimated clustering Z', $p(Z)$ and $p(Z')$ will be the probability distribution for a node belonging to each of the clusters in Z and Z' respectively. The mutual information will hence be computed by the product over all combinations of cluster assignments between Z and Z', as illustrated in figure 3.3.

Mutual information can be utilised to evaluated the similarity of clusterings obtained from synthetic, generated data where the underlying true clustering is known, but can also be used to compare different inferred clusterings. For networks of brain connectivity such clusterings can be obtained either from multiple inferences on the same network, from networks based on rescans of the same subject or from networks based on scans for other subjects. The

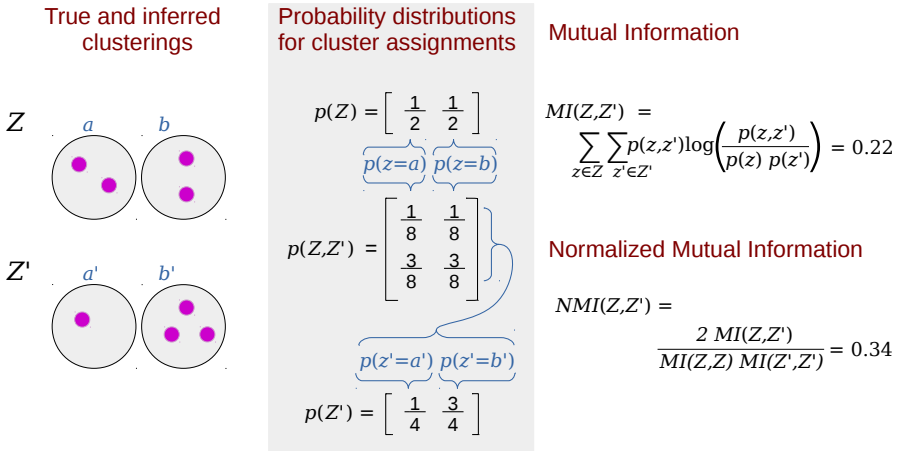


Figure 3.3: Mutual Information. *Example of computing the mutual information (MI) and Normalized Mutual Information (NMI) between two clusterings Z and Z' that each partitions the four nodes into two clusters.*

upper bound for the mutual information score depends on the complexity of the clusterings. In order to get a measure between zero and one, a normalized mutual information can be defined as:

$$NMI(Z, Z') = \frac{2\, MI(Z, Z')}{MI(Z, Z) + MI(Z', Z')}$$

Here a score of zero describes that the two clusterings share no information, while a score of one describes that the two clusterings are identical.

3.1.3.2 Link prediction

In many situations there is no ground truth available. To quantify the performance of the model in such situations, we evaluate the models predictive ability on hold out data. This is done by withholding some the data from the model inference and afterwards evaluate how well the inferred model is capable of predicting this unseen data. When the data consists of just a single network, the held-out data can ideally be obtained by denoting a certain percentage of randomly selected links and non-links in the network as held-out data. During the model inference the held-out links and non-links should ideally be considered as unknown or missing data [Miller et al., 2009]. An even more cautious link-prediction strategy is to simply treat the held-out links as non-existing

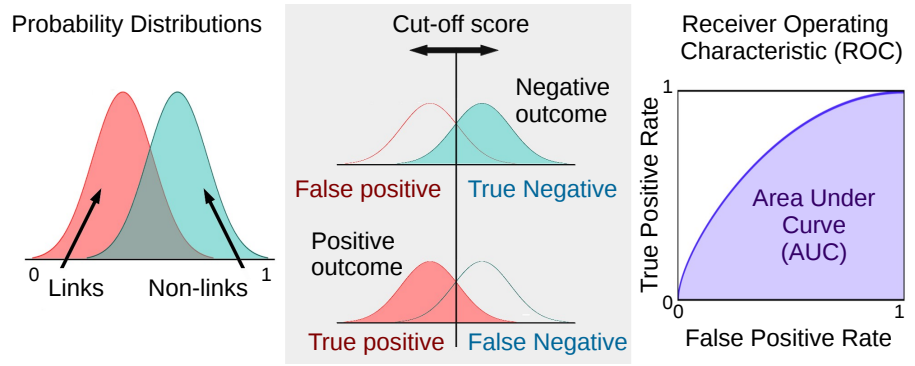


Figure 3.4: Area Under Curve. *Concept for evaluating a models predictive ability by computing the Area Under Score (AUC) of the Receiver Operating Characteristic (ROC) curve. The difference between the overlapping distributions for links and non-links indicates how well the model can separate links from non-links.*

[Liben-Nowell and Kleinberg, 2007, Clauset et al., 2008]. This strategy is more prone to overfitting and might hence easier illustrate whether a model exhibit overfitting issues. If the data consists of multiple networks, one or more entire networks can be designated as held-out data. Probabilities for observing links or non-links in the held out data can hence be computed using the inferred model parameters and compared with the actual data to quantify how well the model separates links and non-links. Figure 3.4 illustrates the concept of evaluating this predictive performance using the Area Under Curve (AUC) of the Receiver Operating Characteristic (ROC) curve. The AUC score will be 1 in the case of perfect predictions (no false positive or false negative) while predictions by random chance yield an AUC score of 0.5.

3.2 Stochastic Blockmodelling

Figure 3.5 illustrates the generative process for creating networks by the stochastic blockmodel. *Firstly*, a partition of all nodes is generated. The finite SBM allows for flexible cluster-sizes by basing the partitioning on a Dirichlet distribution. The number of clusters K is however a parameter of the model that must be defined in advance. In the infinite relational model (IRM) the partitioning is based on a Chinese Restaurant Process (CRP) [Aldous, 1985] which allows a countable infinite number of clusters. *Secondly*, the probabilities for links between and within individual clusters are determined. *Finally*, links

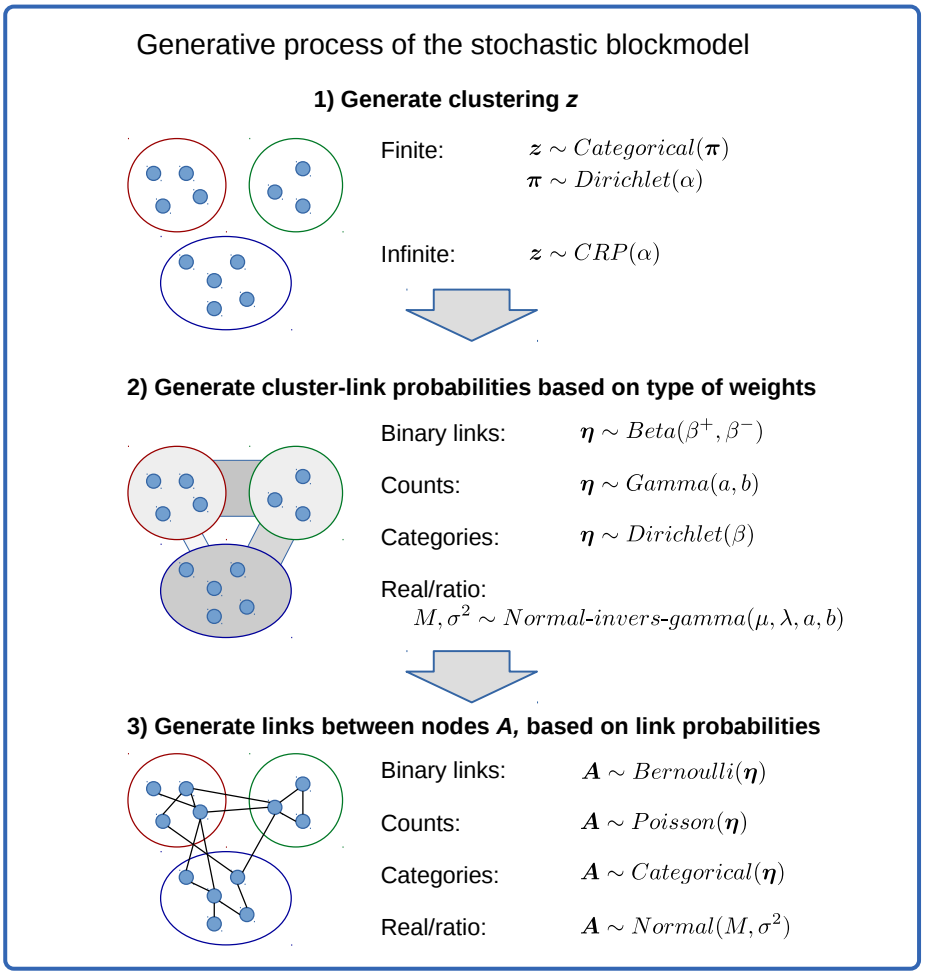


Figure 3.5: The stochastic blockmodel. *The generative process of the stochastic blockmodel. Distributions for link-probabilities between clusters are chosen as conjugate prior distributions for the likelihood p(A), which is chosen according to the wanted type of links in the network.*

between individual nodes are generated based on the probability of observing links between the clusters, creating the assignment matrix $\boldsymbol{A}$ of the network. The stochastic blockmodel is simplified by the assumptions that each node belongs to exactly one cluster, and that the probability of observing interactions between two nodes only depends on the clusters, that the two nodes belong

to[Nowicki and Snijders, 2001].
The stochastic blockmodel is an intuitive and simple statistical model for discovering the latent clustering structure from observed complex network. In practical clustering problems, the links $\boldsymbol{A}$ of a given network are observed and we wish to infer an appropriate clustering, that according to the model would have been likely to have generated the observed links. The distribution by which links between nodes are considered generated should be based on the type of the links that are in the observed network. The distribution for the links between clusters are hence chosen such that it acts as conjugate prior.

3.2.1 Dirichlet-categorical clustering prior

Let $\boldsymbol{\pi} = \{\pi_1, ..., \pi_K\}$ be the probability distribution of any node belonging to any of the K clusters ℓ such that $p(z_i = \ell|\pi) = \pi_K$ and $\sum_{\ell=1}^{K} \pi_k = 1$, where z_i denotes the cluster assignment of node i. To allow for flexible cluster sizes, $\boldsymbol{\pi}$ is based on the Dirichlet distribution:

$$p(\boldsymbol{\pi}|\boldsymbol{\lambda}) = \frac{1}{B(\boldsymbol{\lambda})} \prod_{\ell=1}^{K} \pi_l^{\lambda_\ell - 1} \quad , \quad \text{where} \quad B(\boldsymbol{\lambda}) = \frac{\prod_{\ell=1}^{K} \Gamma(\lambda_\ell)}{\Gamma(\sum_{\ell=1}^{K} \lambda_\ell)}. \tag{3.4}$$

Here $B()$ is the multivariate beta function with $\Gamma(a) = (a-1)!$ being the gamma function.

As no cluster is preferred in advance, we impose equal concentration parameters for all clusters (see [Schmidt and Mørup, 2013] for details). We define the parameter $\alpha = \sum_{k=1}^{K} \lambda_k$ such that $\frac{\alpha}{K} = \lambda_1 = ... = \lambda_K$. The joint prior over $\boldsymbol{z}$ and $\boldsymbol{\pi}$ can be written as:

$$p(\boldsymbol{\pi}, \boldsymbol{z}|\boldsymbol{\lambda}) \;=\; p(\boldsymbol{\pi}|\boldsymbol{\lambda}) \prod_{i=1}^{N} p(z_i|\boldsymbol{\pi}) \;=\; \frac{1}{B(\boldsymbol{\lambda})} \prod_{\ell=1}^{K} \pi_l^{n_l + \lambda_\ell - 1}, \tag{3.5}$$

where n_ℓ is the number of nodes in cluster ℓ and N is the total number of nodes. Due to the conjugacy of the categorical and Dirichlet distributions, $\boldsymbol{\pi}$ can be marginalized, resulting in the following effective prior:

$$p(\boldsymbol{z}|\alpha) = \int p(\boldsymbol{\pi}, \boldsymbol{z}|\alpha)\, d\pi = \frac{\Gamma(\alpha)}{\Gamma(\alpha + N)} \prod_{\ell=1}^{K} \frac{\frac{\alpha}{K} + n_\ell}{\Gamma(\frac{\alpha}{K})} \tag{3.6}$$

This is a multivariate Pólya distribution and is used as prior for the clustering in the finite SBM, relying on the single hyperparameter α and the number of clusters K.

3.2.2 CRP clustering prior

If we do not want to specify the number of clusters in advance, the Chinese restaurant process (CRP) [Aldous, 1985] can be used as prior for the clustering. CRP can be described as a iterative procedure where the nodes are partitioned into clusters one at a time, such that the partition of a node only is based on the partition of all previously considered nodes. Even though the process allows for infinitely many clusters, each node assignment will only be based on the probabilities of assigning the node to all currently non-empty clusters as well as the first currently empty cluster [Gershman and Blei, 2012]. The process is illustrated in figure 3.6. A node n_i is placed in a cluster according to the following probabilities:

$$p(z_{n_i} = \ell | z_{n_1}, ...z_{n_{i-1}}, \alpha) = \begin{cases} \frac{\alpha}{i+\alpha-1} & \text{if cluster } \ell \text{ is empty} \\ \frac{m_\ell}{i+\alpha-1} & \text{if cluster } \ell \text{ is not empty} \end{cases} \tag{3.7}$$

Here m_ℓ is the number of nodes in cluster ℓ and i is the current number of considered nodes. Though CRP allows for a possible infinite number of clusters only a finite number will be involved in the process of partitioning the data.

CRP exhibits some important properties. *First*, the cluster assignments are exchangeable. The probability of a given clustering only depends on the number of clusters K and the size of the clusters $m_1, ..., m_K$. It is irrelevant in which order the nodes are placed [Gershman and Blei, 2012]. *Second*, CRP exhibits the rich-get-richer phenomenon. The generative process prefers to populate already well-populated clusters as the probability of placing a node in a cluster ℓ is higher the larger m_ℓ. *Third*, every time a node is placed it becomes less likely that the next node will be placed in an empty cluster. This is due to the rich-get-richer phenomenon, and meets our expectation that in order for there to exist larger clusters, the number of clusters K must be considerably smaller than the number of nodes N. The expected number of clusters after partitioning N nodes is: $\sum_{i=1}^{N} \left(\frac{\alpha}{i+\alpha+1}\right)$, which as a harmonic series grows logarithmically in N. The number of clusters is however influenced directly by the value of the concentration parameter α, as increasing α implies an increased number of clusters.

From 3.7 we can compute the probability for an entire partition $\boldsymbol{z}$ with K clusters over N nodes. To generate $\boldsymbol{z}$ the CRP must have placed a node in a new cluster K times. As each cluster ℓ contains m_ℓ nodes, the CRP must have assigned $m_\ell - 1$ nodes to the cluster when it was non-empty. The probability of $\boldsymbol{z}$ is hence given as:

$$p(\boldsymbol{z}|\alpha) = \frac{\alpha^K (\alpha-1)! \prod_\ell^K (m_\ell - 1)!}{(N+\alpha-1)} = \frac{\alpha^K \Gamma(\alpha) \prod_{\ell=1}^K \Gamma(m_\ell)}{\Gamma(N+\alpha)} \tag{3.8}$$

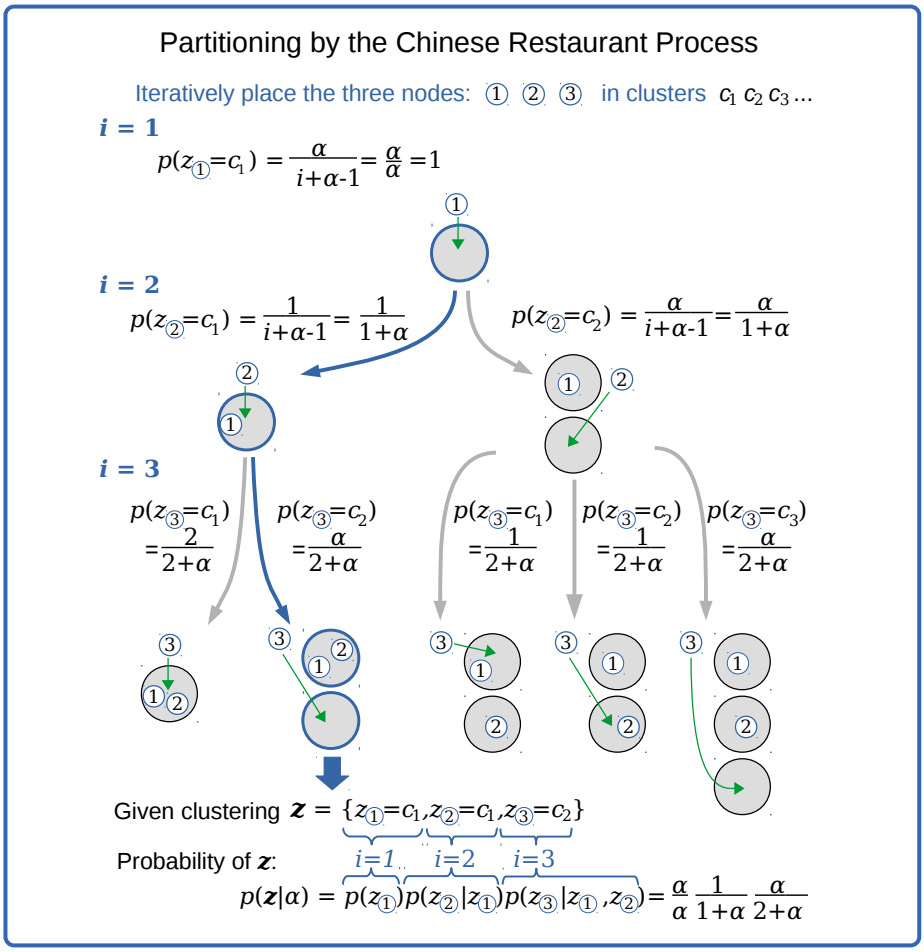


Figure 3.6: The Chinese Restaurant Process. *For three nodes, the figure illustrates the iterative procedure of obtaining a clustering by CRP and shows an example for computing the probability of a given clustering.*

3.2.3 Bernoulli likelihood and Beta prior

Let $\boldsymbol{A}$ represent the binary adjacency matrix of a simple unweighted and undirected network with N nodes. For all pair of nodes i and j, let $A_{ij} = 1$ if there exists a link between the nodes and $A_{ij} = 0$ otherwise.

The Bernoulli distribution is a probability distribution of a single binary random variable x, with outcome $x = 1$ with probability θ and $x = 0$ with probability $1 - \theta$, resulting in the following probability density function:

$$p(x|\theta) = \theta^x(1-\theta)^{1-x}, \quad \text{for } x \in \{0, 1\} \tag{3.9}$$

The probability of observing a link in the graph can hence be set to follow the Bernoulli distribution.

$$A_{ij} \sim Bernoulli(\eta_{z_i z_j}),$$

only depending on the probability of observing links between the clusters z_i and z_j, that the two nodes belong to. For an undirected network with no self-links, $A_{ij} = A_{ji}$ and $A_{ii} = 0$. The probability of observing all links in the network is hence given as:

$$p(\boldsymbol{A}|\boldsymbol{\eta}) = \prod_{i<j}^{N} \eta_{z_i z_j}^{A_{ij}}(1 - \eta z_i z_j)^{1-A_{ij}} = \prod_{l \leq m} \eta_{lm}^{N_{lm}^{+}}(1-\eta_{lm})^{1-N_{lm}^{-}}, \tag{3.10}$$

where the last product is over all pair of clusters. N_{lm}^{+} denotes the total number of links between cluster l and m while N_{lm}^{-} denotes the total number of possible yet non-observed links between the clusters.

The cluster-link probabilities $\boldsymbol{\eta}$ can be set to follow the Beta distribution, which acts as conjugate prior to the Bernoulli likelihood.

$$\eta_{lm} \sim Beta(\beta^{+}, \beta^{-})$$

As link-probabilities between clusters are considered independent, we get the following product:

$$p(\boldsymbol{\eta}|\beta^{+}, \beta^{-}) = \prod_{l \leq m} \frac{\Gamma(\beta^{+} + \beta^{-})}{\Gamma(\beta^{+})\Gamma(\beta^{-})} \eta_{lm}^{\beta^{+}-1}(1-\eta_{lm})^{\beta^{-}-1}, \tag{3.11}$$

where $\Gamma(x) = (x-1)!$ denotes the gamma function. From 3.10 and 3.11, we obtain the following joint distribution:

$$\begin{aligned} p(\boldsymbol{A}, \boldsymbol{z}, \boldsymbol{\eta}|\alpha, \beta^{+}, \beta^{-}) &= p(\boldsymbol{z}|\alpha)\, p(\boldsymbol{A}|\boldsymbol{\eta})\, p(\boldsymbol{\eta}|\beta^{+}, \beta^{-}) \\ &= p(\boldsymbol{z}|\alpha) \prod_{l \leq m} \left(\frac{\Gamma(\beta^{+} + \beta^{-})}{\Gamma(\beta^{+})\Gamma(\beta^{-})} \eta_{lm}^{\beta^{+}+N_{lm}^{+}-1}(1-\eta_{lm})^{\beta^{-}+N_{lm}^{-}-1} \right) \end{aligned} \tag{3.12}$$

Because of the conjugate prior, we can analytically integrate to collapse $\boldsymbol{\eta}$:

$$\begin{aligned} p(\boldsymbol{A}, \boldsymbol{z}|\alpha, \beta^{+}, \beta^{-}) &= \int p(\boldsymbol{A}, \boldsymbol{z}, \boldsymbol{\eta}|\alpha, \beta^{+}, \beta^{-})\, d\boldsymbol{\eta} \\ = \quad p(\boldsymbol{z}|\alpha) \prod_{l \leq m} \left(\frac{\Gamma(\beta^{+} + \beta^{-})}{\Gamma(\beta^{+})\Gamma(\beta^{-})} \int_0^1 \eta_{lm}^{\beta^{+}+N_{lm}^{+}-1}(1-\eta_{lm})^{\beta^{-}+N_{lm}^{-}-1} d\eta_{lm} \right) \end{aligned} \tag{3.13}$$

Using the beta-function defined as:

$$B(\beta^+, \beta^-) = \int_0^1 \eta^{\beta^+ - 1}(1-\eta)^{\beta^- - 1}\, d\eta = \frac{\Gamma(\beta^+)\Gamma(\beta^-)}{\Gamma(\beta^+ + \beta^-)},$$

we can write 3.13 as:

$$p(\boldsymbol{A}, \boldsymbol{z}|\beta^+, \beta^-, \alpha) = p(\boldsymbol{z}|\alpha) \prod_{l \leq m} \frac{B(N_{lm}^+ + \beta^+, N_{lm}^- + \beta^-)}{B(\beta^+, \beta^-)} \tag{3.14}$$

For $p(\boldsymbol{z}|\alpha)$ we can either choose the CRP prior (3.8) to get an infinite model (IRM) or chose the Dirichlet-categorical prior (3.6) to get a finite stochastic blockmodel. For a finite model the number of clusters K can be defined in advance or determined by various model selection strategies. In either way, expression 3.14 can be used with Bayes' theorem to obtain the posterior distribution $p(\boldsymbol{z})$ for the clustering, as:

$$p(\boldsymbol{z}|\boldsymbol{A}, \beta^+, \beta^-, \alpha) = \frac{p(\boldsymbol{A}, \boldsymbol{z}|\beta^+, \beta^-, \alpha)}{\sum_{\boldsymbol{z}'} p(\boldsymbol{A}, \boldsymbol{z}'|\beta^+, \beta^-, \alpha)} \tag{3.15}$$

This expression can conceptually be used to identify the clusterings that are probable to have been generated by the model based on a given network $\boldsymbol{A}$. It can hence be used to solve the clustering problem by identifying appropriate clusterings for an observed network. In a practical setting this approach is not feasible due to the number of combinations of possible clusterings. Instead we can approximate the correct distribution by MCMC simulations as presented in section 3.3. Here we utilize that Bayes' theorem can be used to obtain the following conditional posterior distribution for the cluster assignment of a single node i, given the assignments of all other nodes:

$$p(z_i = l|\boldsymbol{A}, \boldsymbol{z}_{\setminus i}, \beta^+, \beta^-, \alpha) = \frac{p(\boldsymbol{A}, \boldsymbol{z}_{\setminus i}, z_i = l|\beta^+, \beta^-, \alpha)}{\sum_m p(\boldsymbol{A}, \boldsymbol{z}_{\setminus i}, z_i = m|\beta^+, \beta^-, \alpha)}, \tag{3.16}$$

where $\boldsymbol{z}_{\setminus i}$ denotes the assignments of all nodes except node i.

3.2.3.1 Missing data

The stochastic block model can intuitively ignore corrupted or missing data in the network. In practice this simply constitutes to leaving out the terms in the likelihood that involve the missing links. A single network can be split into training and hold-out data, by simply designating part of the network as missing (both designating links and non-links according to their distribution). For modelling brain connectivity we prefer to use multiple networks, either from

rescans of the same subject or based on a population of subjects. In this case some entire networks can be used for the inference while other networks can be used as held-out test data.

3.2.3.2 Multiple and population networks

Many clustering problems resolve around finding a single partitioning over multiple networks for the same set of nodes. For neuroimaging this is often the case when modelling data that contains rescans of the same subject or scans for multiple subjects.

The stochastic blockmodel can be specified to infer a single clustering from multiple assignment matrices of the same size. Let $\boldsymbol{A} = \{\boldsymbol{A}^{(1)}, ..., \boldsymbol{A}^{(S)}\}$ represent the set of S assignment matrices. The probability of observing links can be considered independent for each network, such that:

$$A_{ij}^{(s)} \sim Bernoulli(\eta_{z_i z_j}^{(s)}), \tag{3.17}$$

$$\eta_{lm}^{(s)} \sim Beta(\beta^+, \beta^-) \tag{3.18}$$

This effectively replaces the likelihood in expression 3.14 with the product over all networks, resulting in the following joint distribution:

$$p(\boldsymbol{A}, \boldsymbol{z}|\beta^+, \beta^-, \alpha) = p(\boldsymbol{z}|\alpha) \prod_{s=1}^{S} \prod_{l \leq m} \frac{B(N_{lm}^{(s)+} + \beta^+, N_{lm}^{(s)-} + \beta^-)}{B(\beta^+, \beta^-)} \tag{3.19}$$

This approach has previously been explored and proven useful for modelling fMRI data [Mørup et al., 2010]. In Paper F we use this approach to jointly model networks of functional and structural connectivity.

Another approach is to construct a single population graph by aggregating the individual assignment matrices $\boldsymbol{A} = \boldsymbol{A}^{(1)} + ... + \boldsymbol{A}^{(S)}$. This effectively constructs a single weighted graph, where the weight between two nodes i and j represents the number of networks that has a link between i and j; such that $A_{ij} = A_{ij}^{(1)} + ... + A_{ij}^{(S)}$. This approach assumes the same link-probabilities exist across subjects and yields the following joint distribution:

$$p(\boldsymbol{A}, \boldsymbol{z}|\beta^+, \beta^-, \alpha) = p(\boldsymbol{z}|\alpha) \prod_{l \leq m} \frac{B(\sum_s N_{lm}^{(s)+} + \beta^+, \sum_s N_{lm}^{(s)-} + \beta^-)}{B(\beta^+, \beta^-)} \tag{3.20}$$

While the first approach allows the model more flexibility, it is more computationally intensive than the second approach. We have utilized the aggregation approach for large scale modelling of brain connectivity for populations of various number of subjects in Paper E.

3.2.4 Directed and weighted networks

In the practical work of modelling large networks of brain connectivity we have only considered undirected binary networks, and will hence focus our discussions to these types of networks. However it is kept in mind that the same model types intuitively can be applied for other network topologies.

Directed networks can be modelled such that the link probabilities depends on the direction of the link. This gives that the cluster link probability $\eta_{\ell m}$ differs from $\eta_{m\ell}$. Effectively this leaves the model fairly unchanged, except that the likelihood in equation 3.14 will no longer be given by the product over all pairs of clusters, but by the product over all ordered pairs of clusters:

$$p(\boldsymbol{A}, \boldsymbol{z}|\beta^+, \beta^-, \alpha) = p(\boldsymbol{z}|\alpha) \prod_{l,m} \frac{B(N_{lm}^+ + \beta^+, N_{lm}^- + \beta^-)}{B(\beta^+, \beta^-)}. \tag{3.21}$$

In the included work we have solely modelled brain connectivity as undirected binary networks, and will hence focus our discussion of inference procedure and software implementation on this type of networks. Defining the generative model for weighted networks is however conceptually similar, as presented in Appendix G. For weighted networks the generative model must be based distributions appropriate for the type of weights in the network. When links conceptually can be considered integer counts, the Poisson distribution is an intuitive choice for the likelihood for $\boldsymbol{A}$, with the gamma function acting as conjugate prior for $\boldsymbol{\eta}$ [Mørup and Schmidt, 2012]. When the links define discrete categories, a categorical distribution can be utilized with a conjugate Dirichlet prior [Mørup et al., 2014]. When weights are represented by continuous real values the normal distribution is suitable [Herlau et al., 2012]. In Appendix G.4 the generative model is defined when both mean and variance of the normal distribution are considered dependent and unknown with a normal-inverse-gamma distribution acting as conjugate prior.

3.3 MCMC inference procedures

When drawing a sample from the generative model, it will be in the form of a network $\boldsymbol{A}$ that is shaped depending on the model parameters $\boldsymbol{z}$ and $\boldsymbol{\eta}$ which in turn depends on the set of hyper-parameters. In the practical clustering problem we are however presented with an observed network $\boldsymbol{A}$ and wish to infer the model parameters based on $\boldsymbol{A}$. In the presented stochastic blockmodels, the link-probabilities $\boldsymbol{\eta}$ between clusters was analytically marginalized. In this

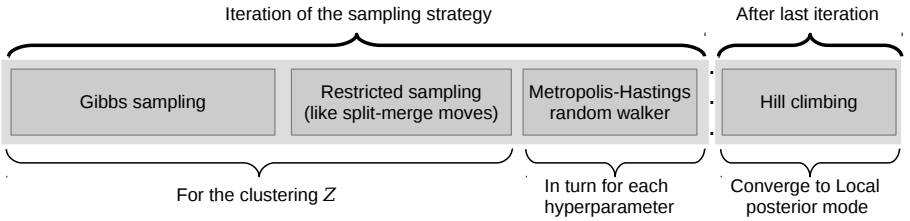


Figure 3.7: Elements of the sampling strategy. *The full sampling strategy consisting of independent samplers for the clustering and hyperparameters. The sampling procedure can be followed by hill climbing optimization to reach the local posterior mode. In Paper C the hill climbing was utilized in order to explore the posterior landscape and relation between network size and number of local modes.*

case the inference will only be concerned with the clustering $\boldsymbol{z}$ and the hyperparameters.

We use an MCMC sampling strategy to infer the model parameters and sample from the posterior distribution. This strategy consists of a sequence of independent MCMC methods, that individually are concerned with proposing changes for just a single model parameter. The sampling strategy is illustrated in figure 3.7.

Each iteration of the sampling strategy consists of sequentially applying all the MCMC methods. To infer the clustering we use a combination of sampling over all nodes and restricted to a subset of nodes. In the full Gibbs sampling each node is in turn proposed to be assigned to other clusters, while the restricted samplers only considers the assignment for nodes within randomly selected clusters. The hyper-parameters are in turn sampled using a Metropolis-Hastings procedure, where new proposals are randomly drawn from a Gaussian distribution, centred at the current value of the parameter.
In some experiments, we wish to optimize the inferred clustering towards the local posterior mode. This is achieved using a hill climbing procedure that reassigns each node to the cluster providing the highest posterior gain. The hill climbing procedure is repeated until no posterior gain can be obtained for any node.

3.3.1 Gibbs sampling

The Gibbs sampler iteratively updates the cluster assignment for each node i. This is done according to the probability of i belonging to the individual clusters, as determined by the conditional distribution of the cluster assignment for i given the cluster assignments for all the other nodes (equation 3.16).

Using Gibbs sampling to sample from this conditional posterior, the node assignment z_i for each node i is in turn processed under the assumption that all other node assignments are fixed and i is ignored. Computing the change in posterior by assigning i to each of the clusters (and a new cluster for IRM) gives a categorical distribution over partitions from which a new cluster assignment for i can be drawn [Schmidt and Mørup, 2013].

From equation 3.14 the change in likelihood of assigning node i to cluster k can be found as:

$$\prod_{\ell} \frac{B(N_{k\ell}^{+\backslash i} + r_{i\ell} + \beta^{+}, N_{k\ell}^{-\backslash i} + n_{\ell}^{\backslash i} - r_{i\ell} + \beta^{-})}{B(N_{k\ell}^{+\backslash i} + \beta^{+}, N_{k\ell}^{-\backslash i} + \beta^{-})}, \tag{3.22}$$

where $n_{\ell}^{\backslash i}$ is the number of nodes in cluster ℓ ignoring node i, $N_{k\ell}^{+\backslash i}$ and $N_{k\ell}^{-\backslash i}$ are the count statistics for links and non-links between cluster k and ℓ, ignoring any links involving node i, while $r_{i\ell}$ is the number of links from i to all nodes in cluster ℓ. The effective change in the prior using CRP is $n_{k}^{\backslash i}$ when k is not empty and α when k is empty. Using the finite prior in equation 3.6, the change becomes $n_{k}^{\backslash i} + \frac{\alpha}{K}$ where K is the number of clusters.

One major issue with Gibbs sampling is mixing over the posterior distribution. For the Markov chain to move between any two well-supported configurations s_a and s_b, the Gibbs sampling must go through a series of intermediate configurations $s_a, s_{a+1}, ..., s_{b-1}, s_b$, that each differs by moving just a single node from one cluster to another. If there exists no path of well-supported configurations between s_a and s_b, the entire scheme might mix poorly [Griffin and Holmes, 2010]. In a previous study, we observed that the mixing ability of Gibbs sampling in the Infinite Relational Model is heavily influenced by the network size and complexity. When applied on the averaged network of brain-connectivity with 998 nodes [Hagmann et al., 2008], we found that with Gibbs sampling alone the model failed to mix over the posterior distribution - even after millions of Gibbs sweeps [Albers et al., 2013].

3.3.2 Split-merge sampling

One proposed way for improving the mixing ability is to supplement the Gibbs sampling with split-merge moves [Jain and Neal, 2004], where multiple nodes are potentially repartitioned between configurations instead of moving a single node at a time. Here new configurations are proposed by either splitting an existing cluster into two or merging two existing clusters into one. The proposals are accepted or rejected according to the Metropolis-Hastings acceptance probability. In order to obtain proposals that are more likely to be accepted, restricted Gibbs sampling is used to generate the proposal configurations [Griffin and Holmes, 2010].

The split-merge algorithm is illustrated in figure 3.8. *First*, two nodes i and j are selected at random uniformly. If i and j are assigned to the same cluster ($z_i = z_j$) it is proposed to split the cluster into two. If the nodes are assigned to different clusters, it is proposed to merge the two clusters into one. The procedures for a split and merge proposal are fairly similar in order to ensure detailed balance. *Second*, all nodes S clustered together with either i or j are randomly split into two clusters with i and j placed separated in the two clusters. *Third*, an intermediate launch state $\boldsymbol{z}_{launch}$ is obtained by a sequence of Gibbs sweeps, restricted to the two clusters and the nodes S. From the launch state the final proposal state is obtained and accepted or rejected according to the Metropolis-Hastings acceptance probability:

$$\alpha(\boldsymbol{z}_{proposal}|\boldsymbol{z}) = \min\left[1, \frac{p(\boldsymbol{A}, \boldsymbol{z}_{proposal}|\beta^+, \beta^-, \alpha)q(\boldsymbol{z}|\boldsymbol{z}_{proposal})}{p(\boldsymbol{A}, \boldsymbol{z}|\beta^+, \beta^-, \alpha)q(\boldsymbol{z}_{proposal}|\boldsymbol{z})}\right] \tag{3.23}$$

For a split proposal the proposal state $\boldsymbol{z}_{split}$ is obtained by a final restricted Gibbs sweep. The transition probability is obtained as the product of the individual transition probabilities of moving the nodes from the launch state to the final split configuration. The transition probability of a merge configuration is always 1 as the transition is deterministic.

Split-merge sampling is not applicable for the finite stochastic blockmodel that does not support populating new clusters. Instead we utilize a restricted Gibbs sampling procedure, where two clusters are selected uniformly at random and a series of Gibbs sweeps are performed, restricted to repartitioning the nodes within the two selected clusters.

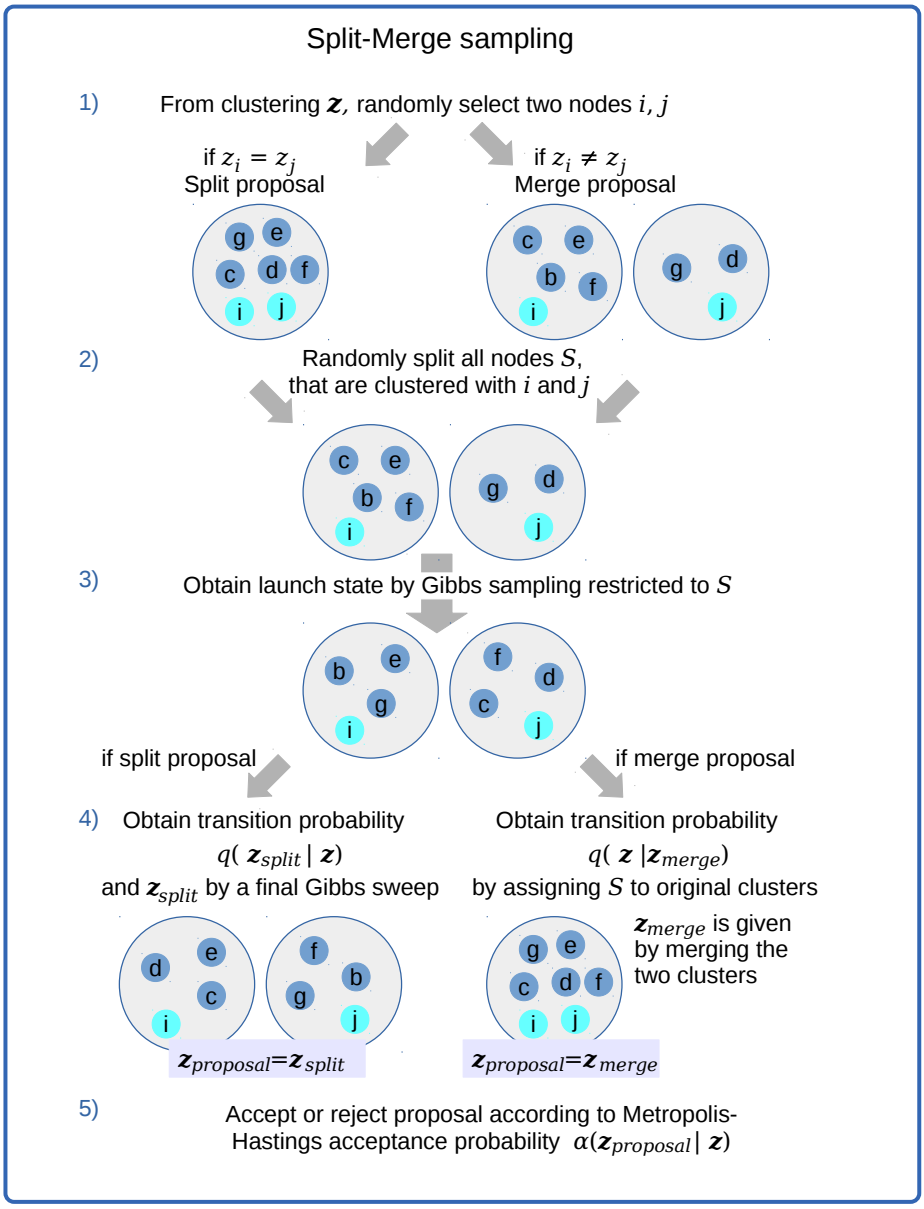


Figure 3.8: The Split-Merge procedure

3.3.3 Sampling of hyperparameters

All hyper-parameters are scalar and can be sampled by the same procedure. We individually sample the hyper-parameters using a Metropolis-Hastings procedure. Here new proposals are drawn from a Gaussian distribution with variance 1 and mean at the current value of the parameter. Proposals are accepted or rejected according to the Metropolis-Hastings acceptance probability.

3.4 Model evaluation strategy

To predict links using an inferred parcellation $\boldsymbol{z}$ the expected link probability between two clusters z_ℓ and z_m can be used as link-prediction score;

$$s_{\ell m} =< \eta_{\ell m} >= \frac{N^+_{\ell m} + \beta^+}{N^+_{\ell m} + N^-_{\ell m} + \beta^+ + \beta^-},$$

which we utilize in Papers D and F. A more conservative strategy that does not rely on the hyperparameters is to base the link-prediction score on the link-density in the observed training network;

$$s_{\ell m} = \frac{N^+_{lm}}{N^+_{lm} + N^-_{lm}}.$$

We use this approach in Paper E where it allows us to compare the predictive performance of different clustering methods. In the paper we also examined several non-parametric link-prediction measures as score for observing links between nodes.

3.5 Implementation and toolbox requirements

Chapter 5 presents the design and implementation for the toolbox that has been developed and used to perform the network modelling in the included work. The toolbox has been designed for both generic usage and high performance as presented in chapter 4. Within the domain of stochastic blockmodelling a list of particular design criteria can be specified in order to ensure that the application is suitable for both high performance and generic usage.

For high performance:

- Define efficient data structures for storing, accessing and modifying the network data, clustering and sufficient statistics (such as the link-counts N^+ and N^-).
- Avoid unnecessary modifications to data structures while the sampling is performed (such as avoiding removing nodes from the clustering and sufficient statistics to use equation 3.22 directly)
- To ensure numeric stability all computations should be performed in the log-domain [Gelman et al., 2014].
- Implement computationally inexpensive evaluations of the logarithm of the Beta function, as this is the key operation to perform the Gibbs sampling when computing in the log-domain (Paper B).

For generic usage:

- Allow the use of different clustering priors.
- Support different network topologies and population graphs.
- Allow a customized setup of the sampling strategy.
- Allow easy implementation of new sampling procedures.
- Allow use of the same sampling implementation across different model implementations.

3.6 Large scale modelling of structural connectivity data

The high performance focus in the implemented toolbox is essential in order to handle the large, complex networks of brain connectivity. The simultaneous generic design allows the toolbox to be intuitively utilized for other problem domains, that might rely on different network types and sampling strategies. Furthermore, the generic design intuitively allows the toolbox to be used as a model selection tool. We have utilized this to identify appropriate number of clusters (Paper H.1, E) and compare population graphs with different number of subjects (Paper E). The advantage of a high performance implementation is that such investigations can be performed on full resolution data within reasonable computation time. In a previous study the generic design has been utilized to examine the influence of different hyper-parameter configurations in the Infinite Relational Model for different sized networks (Paper A). We identified that

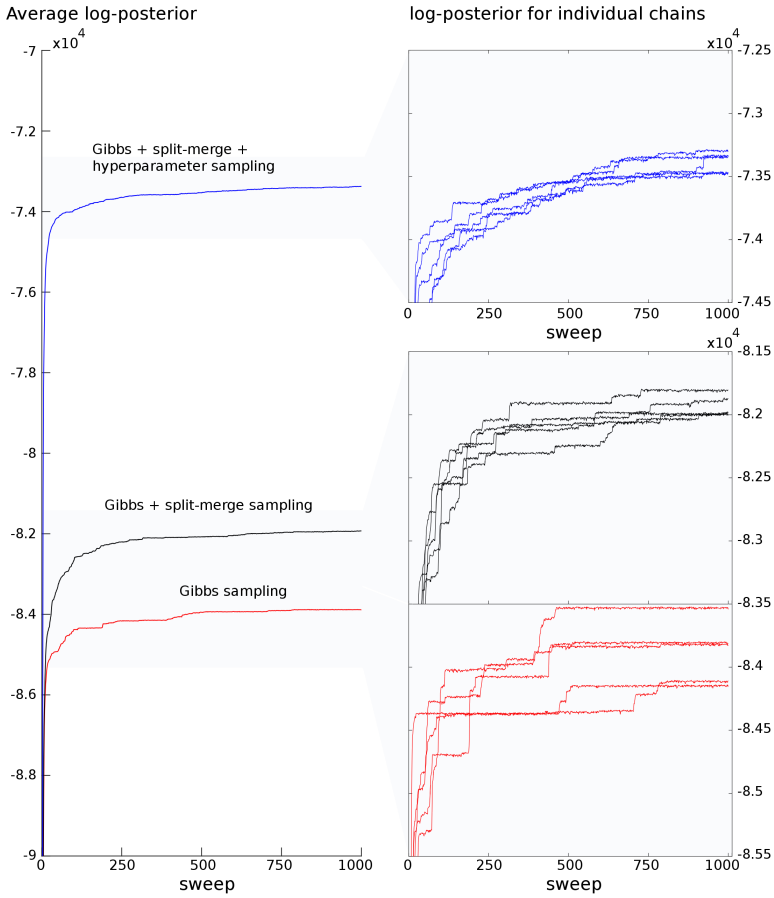


Figure 3.9: Obtained posterior for different sampling strategies. *Logarithm of the posterior value for 1000 samples of the three sampling strategies on the averaged Hagman network of brain connectivity with 998 nodes, using the Infinite Relational Model for binary networks. The sampling strategies are: Gibbs sampling alone, Gibbs with split-merge sampling, and Gibbs with split-merge and hyperparameter sampling. Split-merge sampling is with 10 proposals with 3 restricted Gibbs-sweeps, performed after each Gibbs sweep. The hyperparameters are sampled with 10 random proposals for each parameter after each Gibbs sweep. If not sampled the hyperparameters are fixed at* $\alpha = \beta^+ = \beta^- = 1$. *The average is over five sequences, initial with all nodes in one cluster.*

sampling β^+ and β^- individually outperformed the use of fixed values as well as sampling a single symmetric prior $\beta = \beta^+ = \beta^-$.

The sampling strategy has a significant influence on both model fit and convergence. Figure 3.9 compares different sampling strategies for IRM on the averaged network of brain connectivity with 998 nodes [Hagmann et al., 2008]. When only using Gibbs sampling the procedure seems to get stuck in local modes, as the individual chains are very separated. In fact we have in a previous study determined that Gibbs sampling alone cannot converge on this network even for million of Gibbs sweeps [Albers et al., 2013]. The chains seem to converge faster when combining Gibbs with split-merge sampling, which results in a higher posterior value of the sampled clusterings in fewer sample sweeps. Including the hyperparameter sampling results in the significantly highest posterior value. This sampling procedure allows the model more complexity, which can be seen from figure 3.10 as it identify way more clusters than the other procedures. Even though this model is more complex, figure 3.10 also indicates that the solutions are more stable as it obtains a higher normalized mutual information between chains.

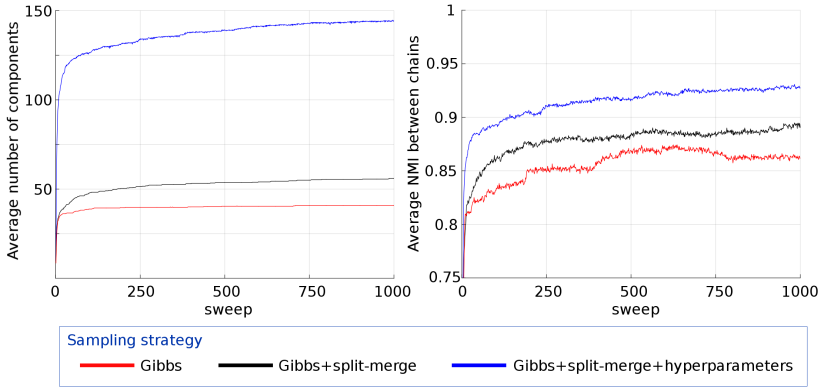


Figure 3.10: Comparing NMI and NOCs for sampling strategies. *Average number of inferred components (NOCs) and Normalized Mutual Information (NMI)for 1000 samples of the three sampling procedures for the averaged Hagman network of brain connectivity with 998 nodes, using the Infinite Relational Model for binary networks. The figures are based on five chains for each sampling strategy as in figure 3.9.*

The predictive performance when using 10 percent of the links as holdout data are compared in figure 3.11.

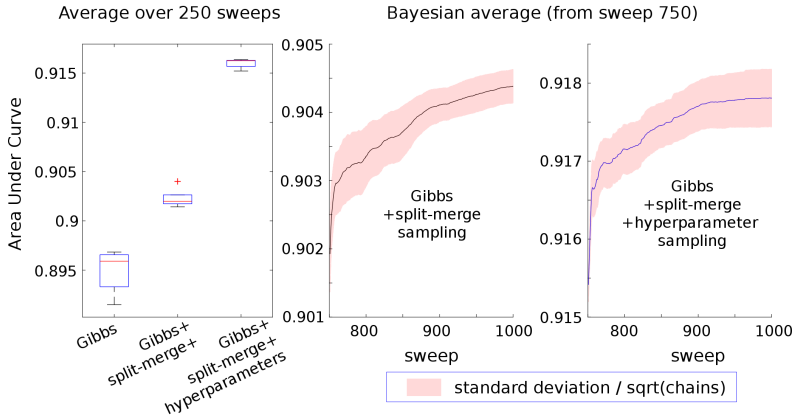


Figure 3.11: AUC for different sampling strategies. *Area Under Curve (AUC) for 1000 samples of the three sampling strategies for the averaged Hagman network of brain connectivity with 998 nodes, using the Infinite Relational Model as in figure 3.9. The boxplot shows the average over the last 250 sweeps, while the Bayesian average at sweep s is computed on the average clustering for sweep 750 to s. The AUC is evaluated using 10 percent of the links as hold out data and are averaged over five chains, all initialized with all nodes in one cluster.*

The added complexity can however make the model more prone to over-fitting to the training data. This is the case in figure 3.12, showing IRM on a graph for a single subject from the HCP data, where the performance is evaluated on predicting graphs for other subjects.

In Paper E we resorted to use the finite stochastic blockmodel (with fixed numbers of clusters). This allowed us to avoid over fitting issues when utilizing the hyper-parameter sampling. Furthermore it allowed for a fair comparison of the predictive performance when modelling with different population sizes, using other sampling procedures and comparing to fixed sized anatomical atlases.

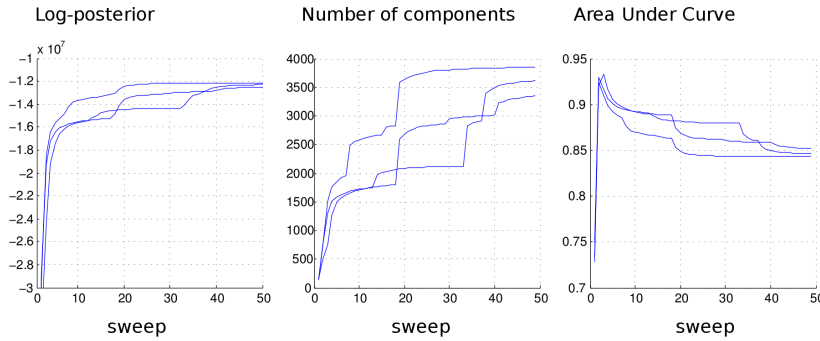


Figure 3.12: IRM on a single HCP subject. *Showing the log-posterior, Area Under Curve and number of inferred components for the first 50 sweeps of IRM. The sampling procedure uses Gibbs, split-merge, and hyperparameter sampling similar to figure 3.9. The figure shows results for three individual chains of the inference based on the same single HCP subject with 59,412 nodes. The AUC is computed and averaged over 6 different test subjects.*

Chapter 4

Statistical computing for Bayesian block modelling

The objective of this chapter is to identify the paradigms for designing and implementing computational tools for performing complex Bayesian modelling on large scaled networks, and describe how the design requirements for our implementation differ from existing tools. A simple overview of the design paradigm for the implemented toolbox is presented, while the realisation of the design and the particular implementation details are discussed in the next chapter.

4.1 Bayesian computing

Since the early pioneering in computer science, computer technology has provided an interesting opportunity for statistical research and experimentation [Von Neumann, 1945]. Since then the entire field of scientific computing and high performance computing has emerged. It has been evolved and shaped by the rapid development of new technologies, computer architectures and usage of systems [Strohmaier et al., 2005]. Since the late 1980s the field of Bayesian computing has emerged. Facilitated by the development in computational tools and necessitated by the increasingly complex scientific research, the Bayesian

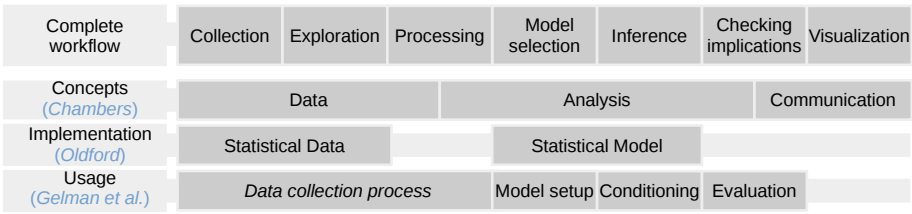


Figure 4.1: *Elements of statistical software, related to the modelling workflow.*

approach blossoms once again, and in certain fields has begun to dominate statistical research [Brooks, 2003].

Today hardware and software solutions play crucial roles in most aspects of the practical workflow of statistical investigations; from data acquisition through the entire modelling process to communication of results. For Bayesian modelling, dedicated computational tools allow scientists to investigate and visualize complex data and use computational intensive procedures (such as Monte Carlo simulations) otherwise impossible.

Figure 4.1 shows three ways of defining and designing software for use in statistical investigations, representing a *conceptual*, a *practical* and an *implementational* approach. Defining the concepts that statistical software can be used for, Chambers separates the data analysis project into three aspects [Chambers, 2000]; to organize, visualize and analyse data. As a software engineering problem, Oldford describes how concepts of statistical modelling can be represented in software [Oldford, 1990]. For each concept there can individually be defined an appropriate software model, that should be based on both the properties of the original statistical concept as well as computational convenience. Oldford notices that relationships between the concepts can and should also be modelled in software, preferably using an object oriented approach [Oldford, 1988]. Our implementation is to be used as a practical tool for modelling already processed data by Bayesian inference. For the practical approach Gelman et al. defines the three steps of Bayesian data analysis [Gelman et al., 2014]; define a full probability model, compute the posterior conditioned on observed data, and evaluate model fit and implications.

4.2 Software for Bayesian inference

Today many software tools exist for performing Bayesian inference, with prominent examples being *Stan*, *BUGS* and *PyMC*.

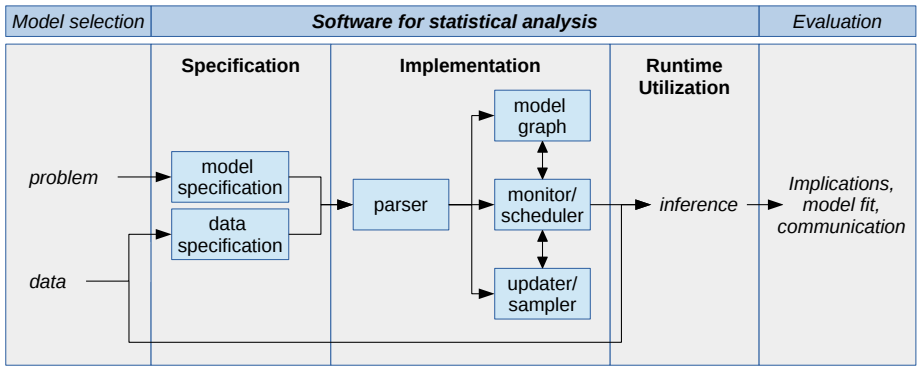


Figure 4.2: Design of software for statistical modelling, *related to the modelling workflow. The conceptual figure is inspired by the design of BUGS [Lunn et al., 2000].*

Stan is a program for general Bayesian analysis, conceived at Columbia University [Stan Development Team, 2012]. It has been continuously expanded, but originally provided a generic framework for automatically applying the Hamiltonian Monte Carlo simulation algorithm [Duane et al., 1987, Neal, 2011] to user-defined Bayesian models. Popular implementations of BUGS including WinBUGS and OpenBUGS [Lunn et al., 2000, Lunn et al., 2009] allows the user to generically specify a statistical model as relations between variables, after which the program automatically can determine and apply an appropriate Gibbs sampling strategy. The initial motivation for PyMC was to make MCMC inference accessible to a broader audience by generalising the process of building Metropolis-Hastings samplers [Patil et al., 2010]. PyMC is developed as a Python module, which allows it to intuitively be incorporated into larger modelling frameworks.

The key strength of these tools are that the user generically is allowed to specify the model and starting values in a high-level language after which the inference procedure automatically is implemented for the resulting posterior distribution. The key elements for designing and using such an application is shown in Figure 4.2. Based on the particular statistical problem, the user specifies the model in a generic model description language. The application contains a parser that can read this language and automatically parse it into an implementation of the full probability model, which can be defined by three software concepts: The *model graph* constitutes an internal representation of the statistical model, after the model specification has been parsed into a graph of interconnected distributions. An *updater* performs MCMC sampling by computing new values for a node in the model graph. Appropriate updaters are coupled to different distri-

butions, supervised by a *monitor* system that schedules the MCMC sampling procedure and stores results in runtime during the inference.

Such generic tools often rely on pattern recognition to simplify the model graph by identifying structures that can be swapped with pre-implemented and performance optimized components. For instance, the performance in Stan is vastly improved by dedicated implementations for both key distributions (including the binomial, gamma, Poisson and normal distributions) and for evaluations of gradients of key statistical expressions [Gelman et al., 2014]. The generic usage of the program is ensured by Stan utilizing automatic analytic differentiation for expressions not known in advance. The user can hence freely specify more exotic models at the cost of the inference being performed slower.

4.3 Design paradigm

Our design differs fundamentally from the generic approach. In our design, the user can not simply provide a model specification, but must provide the full model implementation. Not relying on automated parsing, the user must provide the model implementation written in a programming language that can be compiled natively with the application. This causes significantly more complex work and user commitment than simply providing a model specification in a generic model language. The scope of this thesis has never been to develop generic tools for Bayesian inference in general, but to scale up Bayesian blockmodelling with MCMC sampling to computationally handle large, complex networks. In principle the user can define any model, but it must be implemented from scratch every time. We have chosen this approach, as it is a more direct way of achieving the computational efficiency necessary for performing the Bayesian inference on the large networks in question.

Our approach does not automatically extract an implementation and contains no model graph. Instead, each model parameter is defined by a given *type* (such as being a clustering or a real value) that is associated with an *interface*. When implementing the model, the user must implement the interface for each model parameter after which suitable MCMC samplers can be freely coupled to infer the model parameters.

The model implementation isolates the performance critical computations, that are model specific and benefits from being performance optimized. Non-critical functionality are decoupled from the model implementation and can be handled generically. Functionality for handling input and output, setting up sampling procedures, binding samplers to model parameters and parsing user arguments

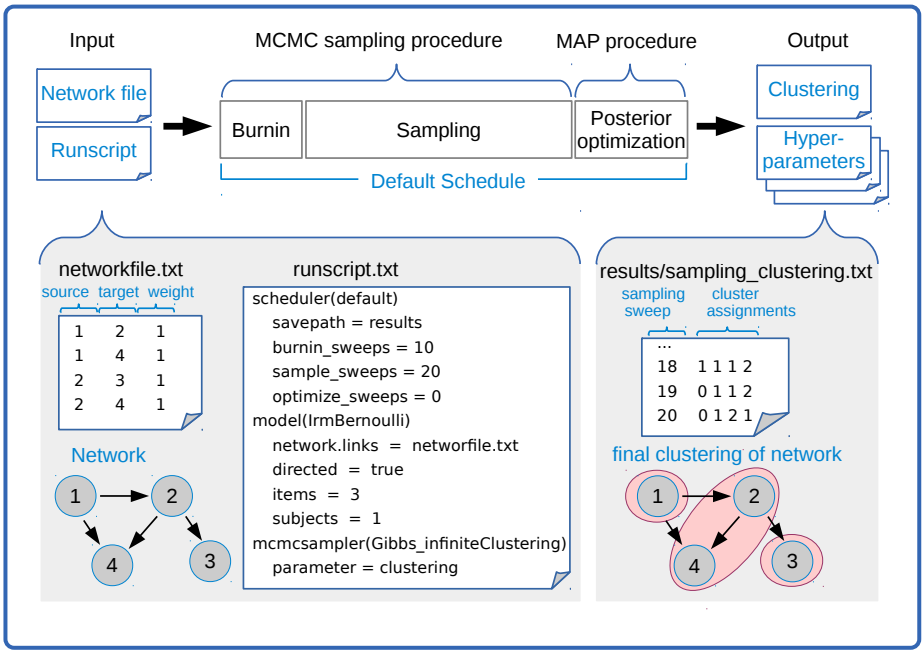


Figure 4.3: Example of specifying an inference procedure in the implemented toolbox. *When models, samplers and schedulers are first implemented, they can be generically utilized. The user provides the compiled application with a file containing the network data and a simple scripting file that specifies the scheduler, the model and how samplers are coupled to model parameters. The program outputs files, containing the inferred values of the sampled model parameters. The example illustrates how to perform IRM on a small network using 30 sweeps of Gibbs sampling.*

are handled generically and requires a low programming effort when creating a new model implementation. Furthermore, when a model implementation is first compiled with the application, it can afterwards be generically utilized. The design allows samplers to be freely coupled to model parameters in runtime, which allows the user to apply different sampling strategies without having to modify or recompile the program.

Figure 4.3 illustrates how the compiled application can be generically utilized. The user provides the application with a file containing the network data and a file that describes what *scheduler*, *model* and *samplers* to use and specifies any optional user arguments for these. The source code is maintained and documented at `https://github.com/kristofferalbers`.

4.4 C++ language features

The implemented tools are written in C++. This is a full-featured, general-purpose and object-oriented programming language, that is widely used in multiple application areas [Dos Reis and Stroustrup, 2011]. Though the C++ language was not particularly designed for numerical and scientific computing [Stroustrup, 2013], the object-oriented approach can often intuitively be used to describe the mathematical abstraction modelled in scientific computing [Budge et al., 1992].
The usage of C++ for implementing research tools is strengthened by the support from a wide range of specialized and cross platform libraries beyond the core ISO standard. Prominent examples are the Boost portable foundation libraries, that extends the core functionality by utilizing the C++ template mechanism [Gerlach and Kneis, 2003], and various libraries providing high-level interfaces for utilizing specialized hardware, such as parallelization on many-core GPU architectures [Demidov et al., 2013].

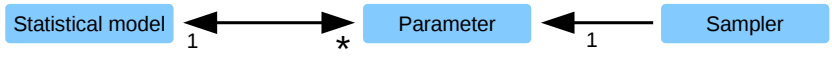
The expressiveness of C++ provides means for *hardware optimizations*; including function inlining, templating and thread level parallelization, *implementational flexibility*; through the notions of abstraction and encapsulation, and *generic usage*; by supporting polymorphism features as virtual functions, dynamic binding and inheritance [Cary et al., 1997]. Of particular importance when designing software for large scale network modelling is that memory management is very performance and hardware optimized; C++ supports efficient memory allocation and memory bandwidth utilization and allows for low-level memory management, direct memory mapping and optimization of CPU cache usage.

4.5 Modular program structure

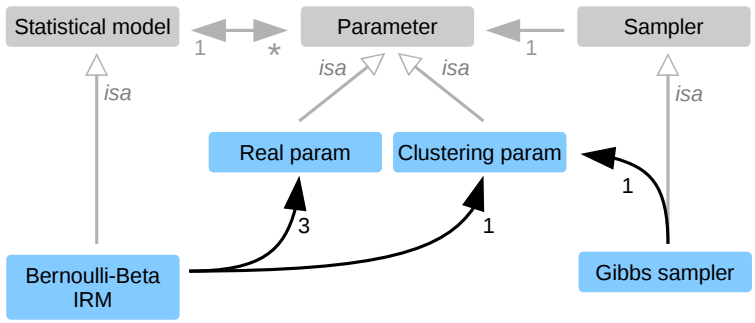
Modular Programming is the principal of splitting the total functionality or service provided by an application into different modules, such that each of these modules provides well-defined services that can be accessed and utilized through well-defined interfaces [Schoett, 1986]. Following the strategy of *separating concerns* the modules must be designed for particular tasks such that they can be implemented and tested separately [Sawitzki, 1996]. Linking the individual modules together will then hopefully provide the wanted functionality of the application in a more generic, expandable and maintainable fashion.

4.5.1 Object oriented design

In an object oriented modular design for our approach to Bayesian block modelling, we can consider the *models*, *parameters* and *samplers* as individual modules. These can be represented by individual classes, structured such that a model object links to a number of parameters, while a given sampler links to one of these parameters:



We want the design to allow the toolbox to easily be extended by implementations of new samplers and models that might rely on various types of parameters. This can simply be achieved by inherit new object types from defined base classes, as in the following example:



Here the class diagram is expanded to contain the classes for representing an instance of the Infinite Relational Model and a Gibbs sampling procedure for the clustering parameter $\boldsymbol{z}$. The class representing the specific model inherits from the Statistical model base class and links to three objects of the Real parameter class (that represent the three hyper-parameters β^+, β^- and α) and further links to one object of the Clustering parameter class that represents the clustering $\boldsymbol{z}$. Both of these parameter classes inherit from the base Parameter class. The Gibbs sampler class inherits from the base Sampler class and links to the single Clustering object, that the sampler is supposed to infer.

A virtual function for computing the posterior value can be defined in the Statistical model base class, implemented in the IRM class and accessed by the Gibbs sampler through the Clustering parameter object. The Gibbs sampler can utilize this virtual function to evaluate the change in posterior if nodes are reassigned, which allows it to carry out the sampling procedure. The sampler and statistical model objects are hence completely decoupled, allowing the same sampler implementation to sample in any model that contains a parameter of the expected type.

Figure 4.4 extends this design by introducing the Scheduler class, being a module that links a number of samplers to a statistical model in order to define a sampling strategy by binding different samplers to the different model parameters. Particular types of sampling procedures or strategies can hence be implemented in classes that inherits from the Scheduler base class. Besides binding samplers to parameters, the scheduler is also responsible for executing the sampling, by in turn invoking each sampler in each sweep of the MCMC sampling. Furthermore the scheduler monitors the sampling procedure and handles user input and output.

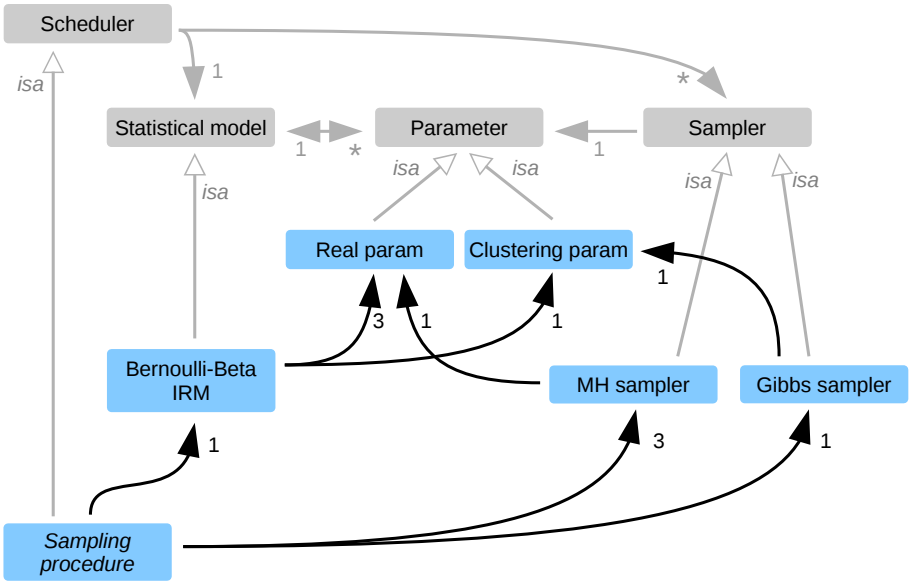


Figure 4.4: Class diagram for stochastic block modelling. *Shows base classes and object instances for a sampling procedure with Gibbs sampling of the clustering parameter and Metropolis-Hastings sampling for each hyper-parameter in IRM.*

4.5.2 High performance

To obtain higher performance, the model object can utilize various assistant data structures for faster evaluations of the posterior value. For the models with Bernoulli likelihood, this includes keeping track of the sufficient statistics describing the number of links and non-links between all pair of clusters. These statistics can be utilized in equation 3.14 to evaluate the entire posterior value and in equation 3.22 to evaluate the posterior change when reassigning nodes.

Whenever parameter values are changed during the sampling procedure, such assistant data structures must also be modified in order to correctly reflect the parameters. The sampler objects can hence no longer simply access and modify the values in the parameter objects directly. The model object must be made aware of any changes to the parameters in order to update or recompute assistant data structures if necessary.

To allow this, we let all functions involving a model parameter being exposed to the sampler through an interface, that defines these functions as virtual functions which are bound and implemented in the model object. The model and sampler object are still completely decoupled, only communicating through a data layer defined by the parameter interface object. The parameter data and model objects are however closely coupled, allowing the use of assistant data structures and model specific optimizations. As the model implementation is hand-coded, the design allows the programmer to fully decide how all this performance critical functionality is implemented, including how the parameter data is stored, accessed and modified as well as how any computations involving the parameters are carried out.

Figure 4.5 illustrates the communication flow between a model and a sampler object, when sampling a single model parameter. The sampler and model only communicates through the virtual functions defined in the parameter interface. The model implement all these functions. Through function calls in the parameter interface, the sampler can request the current value of the parameter and the effect of changing the parameter. Based on this information the sampler object determines whether the value of the parameter must be updated. The function call to inform any change to the parameter value is also in the model object, allowing the model to update any assistant data structures accordingly.

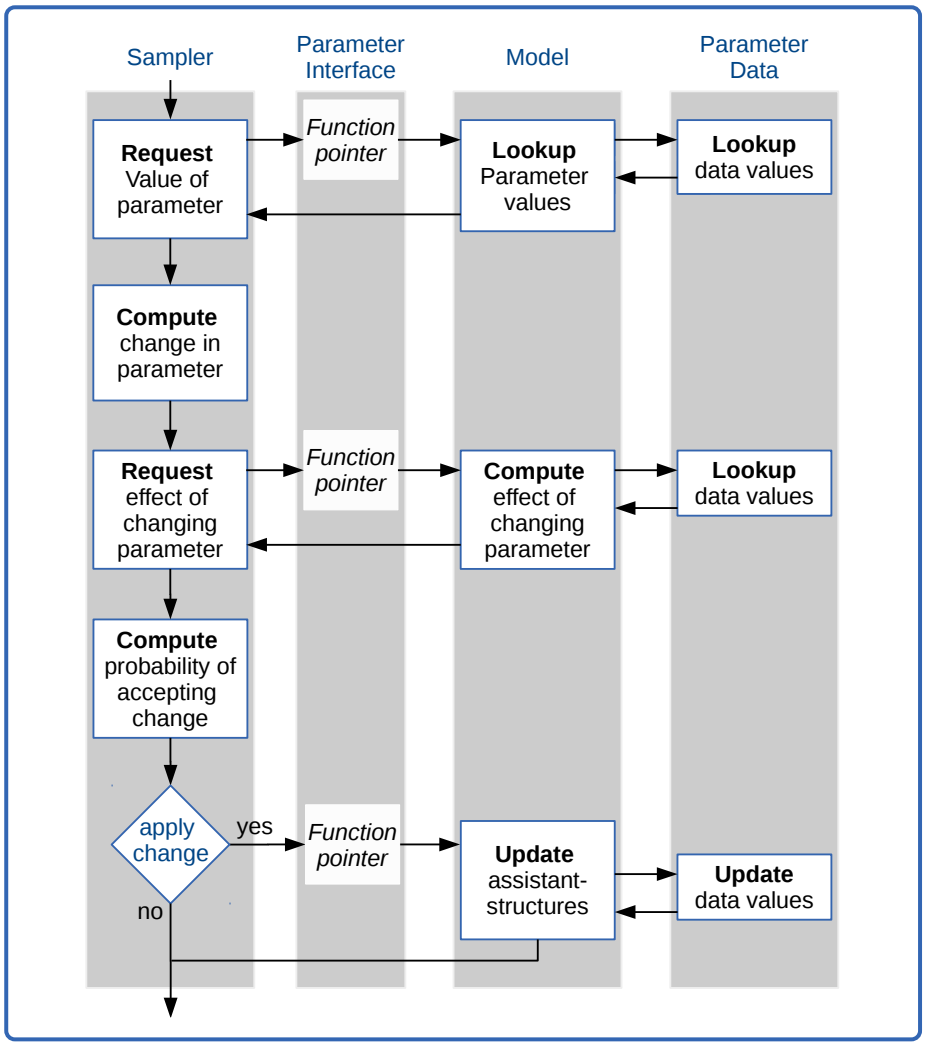


Figure 4.5: Communication flow between sampler and model. *Illustrates the loose coupling between sampler and model, which only communicates through a data layer defined by the Parameter Interface. All functions for accessing and modifying the model parameters are bound and implemented in the model object, exposed to the sampler through the Parameter Interface.*

Chapter 5

Implementation

This chapter presents how the design paradigm has been realised for implementing the toolbox for both generic usage and high performance.

Though the implementation paradigm generalizes to all models, parameters and samplers, it will be discussed by considering the particular implementation for the Infinite Relational Model with the Gibbs sampler for the clustering parameter. Based on this example the dependencies between the program modules that defines samplers, parameters and models are presented.
The data structures for representing network data and clusterings as well as some of the auxiliary data structures and algorithms that can be utilized for improved performance are then presented.

5.1 Program modules

Figure 5.1 shows a class diagram for the relations between the classes necessary for implementing the Infinite Relational Model, a clustering parameter and a Gibbs sampler.

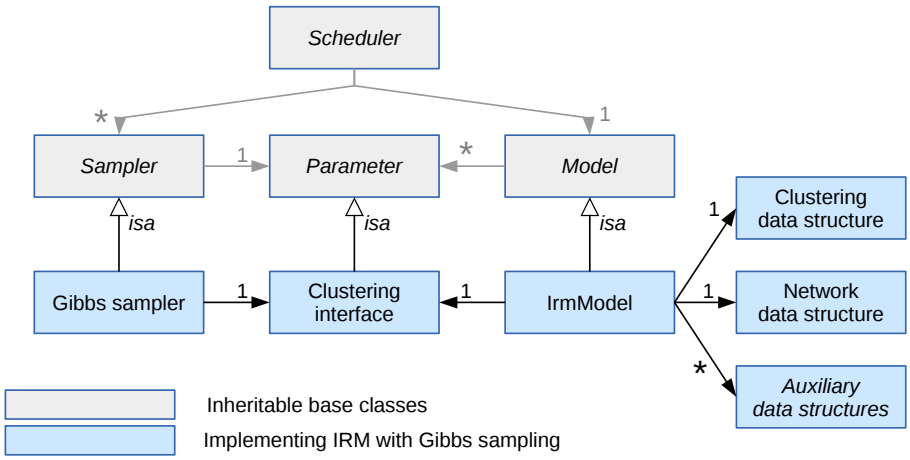


Figure 5.1: Class diagram, showing classes for implementing the Infinite Relational Model, a clustering parameter and a Gibbs sampler.

When performing Gibbs sampling in the Infinite Relational Model, the following three objects are of particular interest:

S is a Gibbs sampler object. It inherits from the Sampler class, which defines the virtual function for invoking the sampling procedure sampler::sample(), such that a scheduler can generically be coupled to different samplers. S contains the algorithms for performing Gibbs sampling on a clustering parameter, which it is aware of as a pointer to the clustering interface Zi.

Zi is an object of the clustering interface class which inherits from the parameter class. Zi links the sampler S to the IRM model object M.

M is an object of the IRM model class. It defines the statistical model and contains data structures for the model parameters including the cluster assignments Z and the network data A. The model object is responsible for performing all computations involving these data structures. To facilitate this, it further contains some auxiliary data structures that can simplify or speed up these computations. The class inherits from the model base class, such that the scheduler generically can couple different model implementations to different sampler implementations. The toolbox implements different model classes for different network topologies and clustering priors, while the network data structure class is templated based on the type of links in the network.

```
function  GibbsSampler::sample()
1  initialize empty vector logP
2  for each node i
3      for each cluster k in clustering Zi                  ┐ Compute change in
4          logP(k) = Zi.logPosteriorRatio(i,k)              ┘ posterior if assigning i to each cluster k
5      end
6      logP(k+1) = Zi.logPosteriorRatio_new(i)              ─ Compute change in posterior if assigning i to a new cluster
7      D = catNormDist(logP)                                ┐
8      r = random()                                         │ Compute normalized categorical distribution for assigning i to clusters based on posterior ratios
9      k = selectCluster(r,D)                               ┘
10     Zi.assignNode(i,k)                                   ─ Assign node i
11 end
```

Figure 5.2: Algorithm for a single Gibbs sweep in IRM. *The procedure relies on the clustering parameter through the clustering interface Zi, that exposes functions for computing the posterior changes when reassigning nodes.*

5.1.1 Sampler implementation

In each Gibbs iteration, the cluster assignment of a given node i is randomly selected based on the categorical distribution, obtained by considering the change in posterior of reassigning the node to all clusters (and a new empty cluster for IRM). Figure 5.2 shows the sample() function that performs a Gibbs sweep within the Gibbs sampler object S. This function relies on a set of functions exposed by the clustering interface object Zi. Similar as presented in the sequence diagram in figure 4.5 the Gibbs sampler utilizes these functions in Zi for the following purposes:

- Obtain information about the current state of the clustering, such as the number of nodes J and current number of clusters K.
- Obtain the effective change in posterior if reassigning a given node i to any of the K clusters or a new empty cluster.
- Instruct any reassignments of nodes based on the sampling.

Within Zi these functions are present only as function pointers that are bound and implemented by member functions in the model object M. This design allows the same sampler implementation to be utilized for any model class that contains

a parameter of the correct type and implements the needed set of functions. A drawback of this design is that the function pointers in the parameter interface object are bound to the member functions in the model object in runtime. This prevents the call structure that involves the function pointers to benefit from inlining and most other compile time optimizations of the program flow.

For seldom called or expensive functions, the added compute time of virtual function calls is negligible. The algorithm in figure 5.2 however contains a loop over all K clusters, with a virtual function call for obtaining the change in posterior of reassigning the node to each of the clusters. For each node, this constitutes K virtual function calls. instead, this loop is moved into the model member function such that a single virtual function call is needed in the sampling algorithm. The model (and parameter interface) instead implements a function that returns a sequential container of computed changes to the log-posterior if reassigning a given node i to the various clusters. The position p within the container object gives the change in log-posterior when assigning the node to cluster p. The function expects parameters to indicate which existing clusters the node must be examined reassigned to. Furthermore, the function expects a Boolean parameter to indicate whether the change in log-posterior of reassigning i to an empty cluster should be included in the container. This design allows $J \times (K + 1)$ virtual function calls to be replaced with J calls in each Gibbs sweep.

5.1.2 Parameter interface implementation

Within the parameter object Zi, the function for computing posterior changes when reassigning nodes is exposed as a function pointer:

```
//definition of function pointer in Zi
std::function
      <vector<double>(size_t,bool,vector<size_t>::iterator)>
                                          logPosteriorRatio;
```

- Return type is a vector of log-posterior changes
- Unsigned int, identifying node *i*
- Boolean to indicate if *i* can also be assigned to a new cluster
- Iterator over a vector of the clusters that *i* can be assigned to
- Name of the function as exposed through the clustering parameter interface

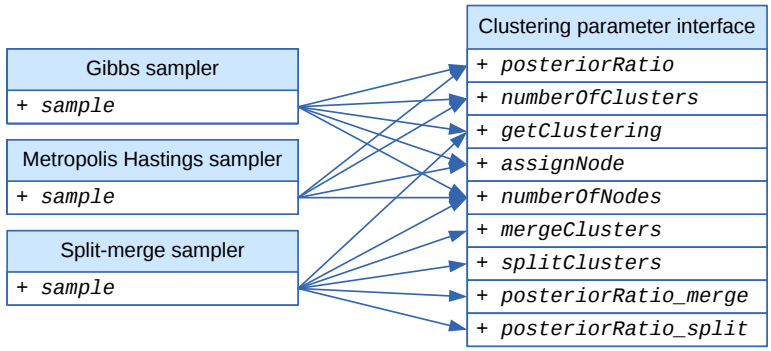


Figure 5.3: Function dependency graph. *Examples of virtual functions, that must be implemented by the model in order to sample a clustering parameter.*

The function pointer is bound to a function within the model object, which performs the actual computations:

```
//definition of the function in the model M
vector<double> logPosteriorRatio_clustering(
       size_t nodeId, bool addNew, vector<size_t>::iterator clusters);

//binding function-pointer to model function
Zi.logPosteriorRatio =
    std::bind(&IrmModel::logPosteriorRatio_clustering,M,_1,_2,_3,);
```

Function in the clustering interface object

Member function implemented in the IrmModel object *M*

Placeholder objects for unbound arguments, defined in the namespace std::placeholders

Being implemented within the model object, the function has access to the clustering data and can benefit from any sufficient and auxiliary statistics maintained by the model in order to improve the runtime performance.

In a similar way, all the parameter functions in Zi can be bound to member functions in the model. Whenever the model implements a parameter P for which it binds all functions that are called by a given sampler, the sampler can be utilized to sample P independent of the type of model.

Figure 5.3 illustrates how different samplers can utilize the same functions. A simple Metropolis-Hastings based procedure that randomly propose new cluster

assignments will need nearly the same set of functions as the Gibbs sampler, while the split-merge sampler needs functions for obtaining the posterior-change and update the clustering for a given merge or split proposal.

5.1.3 Model implementation

The model class is coupled with the network and clustering data as well as any auxiliary data that can be utilized to simplify and speed up computations.

The key operation when supporting the Gibbs sampler is to evaluate the posterior change of reassigning nodes. Recall equation 3.22 for computing the change in likelihood when node i is assigned to cluster k:

$$\prod_{\ell} \frac{B(N_{k\ell}^{+\setminus i} + r_{i\ell}^{+} + \beta^{+}, N_{k\ell}^{-\setminus i} + r_{i\ell}^{-} + \beta^{-})}{B(N_{k\ell}^{+\setminus i} + \beta^{+}, N_{k\ell}^{-\setminus i} + \beta^{-})}, \tag{5.1}$$

where $r_{i\ell}^{-} = n_{\ell}^{\setminus i} - r_{i\ell}^{+}$ is the number of nonlinks from node i to all nodes in cluster ℓ.

Instead of computing the sufficient statistics N^{+} and N^{-} whenever they are needed, they are precomputed and kept updated whenever a node assignment changes, such that they can be reused between Gibbs iterations. Figure 5.4 illustrates how the sufficient statistics can be modified when a node i is ignored and reassigned, by using two computed vectors; $r_i^{+} = [r_{i0}^{+}, ..., r_{iK-1}^{+}]$ and $r_i^{-} = [r_{i0}^{-}, ..., r_{iK-1}^{-}]$, that respectively describes the number of links and nonlinks between i and any nodes in each of the K clusters.

To use equation 5.1 directly, node i can be ignored in the data structures to obtain $N^{+\setminus i}$ and $N^{-\setminus i}$. This can be achieved by subtracting r_i^{+} and r_i^{-} from N^{+} and N^{-} as shown in figure 5.4. While this approach is significantly faster than recomputing $N^{+\setminus i}$ and $N^{-\setminus i}$ whenever they are needed, it has some drawbacks: *First*, it gives a slight computational overhead as the node must be reinserted in the data structures even when it is not reassigned to another cluster. *Second*, computations cannot be performed for different nodes in parallel when the data structures are modified to ignore a single particular node.

Alternatively, $N_{k\ell}^{+\setminus i}$ and $N_{k\ell}^{-\setminus i}$ can be computed from N^{+} and N^{-} without changing the stored data. The computation for each term in 5.1 will hence depend on whether i is currently assigned to the clusters ℓ or k. If this is the case then the links and nonlinks associated with i (given by r_i^{+} and r_i^{-}) must be ignored when computing $N_{k\ell}^{+\setminus i}$ and $N_{k\ell}^{-\setminus i}$. Let c be the current cluster assignment for node i:

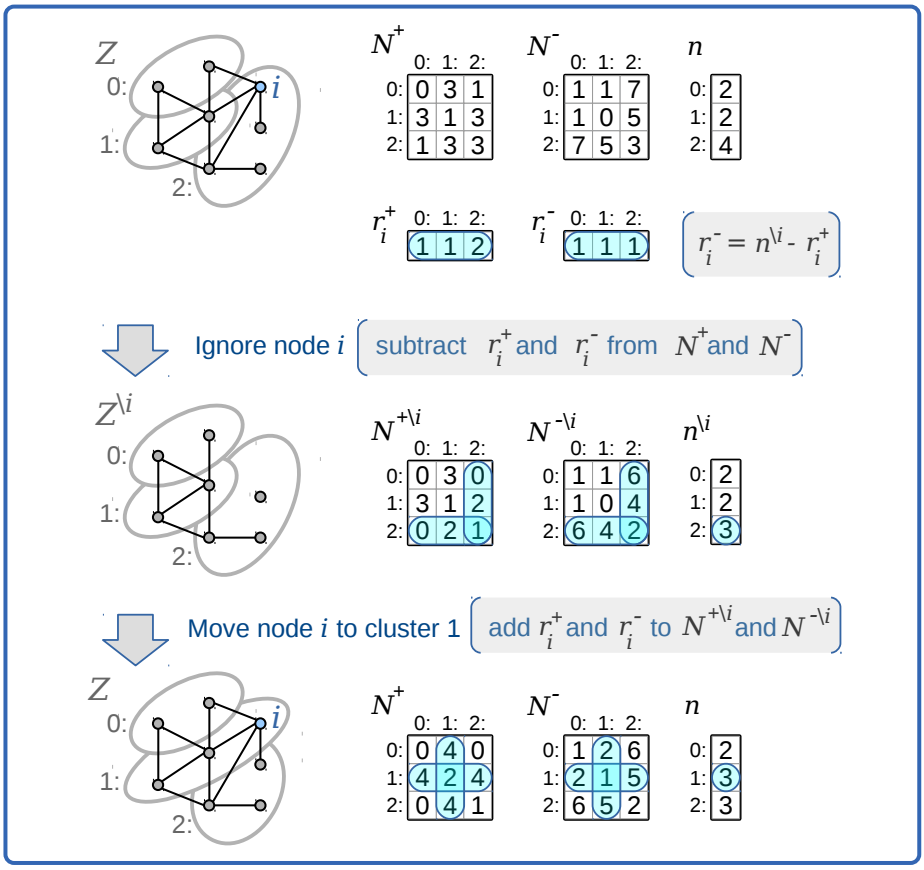


Figure 5.4: Computing sufficient statistics. *Updating N^+,N^-,n when ignoring and reassigning a node, illustrated for an undirected binary network, where the node i is assigned from cluster 2 to cluster 1.*

- If i is not assigned to either of the clusters the stored values can be used directly; $N_{k\ell}^{+\backslash i} = N_{k\ell}^{+}$ and $N_{k\ell}^{-\backslash i} = N_{k\ell}^{-}$.
- If i is assigned to one of the clusters $c = \ell$, the links and non-links associated with i between cluster k and ℓ must be ignored; $N_{k\ell}^{+\backslash i} = N_{k\ell}^{+} - r_{ik}^{+}$ and $N_{k\ell}^{-\backslash i} = N_{k\ell}^{-} - r_{ik}^{-}$.

This approach allows the data structures for the sufficient statistics to remain unmodified while the Gibbs sampling is performed. Only when a node is actually

reassigned they are updated as in figure 5.4. The perceived number of nodes when ignoring node i also depends on the cluster, such that $n_\ell^{\setminus i} = n_\ell - 1$ when i is located in cluster ℓ and $n_\ell^{\setminus i} = n_\ell$ otherwise.

For directed networks N^+ and N^- will not be symmetric. When modelling links in both directions the likelihood is given by the product over all pairs of clusters instead of all ordered pairs, as presented in section 3.2.4. When computing the posterior change of reassigning a node, equation 5.1 must likewise be computed in both directions. The only exception being terms where $k = \ell$ as directionality is not modelled within clusters.

Let $r_{i\ell}^+$ define the number of links *from* node i to any node in cluster ℓ while $\widehat{r}_{i\ell}^+$ defines the number of links *to* node i from all nodes in ℓ. Similarly $r_{i\ell}^-$ and $\widehat{r}_{i\ell}^+$ respectively represent the number of nonlinks from and to node i linked to cluster ℓ. Let c be the current cluster assignment for node i:

- If i is not assigned to either of the clusters the stored values can be used directly:
$$N_{k\ell}^{+\setminus i} = N_{k\ell}^+ \quad , \quad N_{k\ell}^{-\setminus i} = N_{k\ell}^-$$
$$N_{\ell k}^{+\setminus i} = N_{\ell k}^+ \quad , \quad N_{\ell k}^{+\setminus i} = N_{\ell k}^+$$
- If i is assigned to one of the clusters $c = \ell$, the links from and to i must be ignored:
$$N_{k\ell}^{+\setminus i} = N_{kc}^+ - \widehat{r}_{ik}^+ \quad , \quad N_{k\ell}^{-\setminus i} = N_{kc}^- - \widehat{r}_{ik}^-$$
$$N_{\ell k}^{+\setminus i} = N_{ck}^+ - r_{ik}^+ \quad , \quad N_{\ell k}^{-\setminus i} = N_{ck}^- - r_{ik}^-$$
- Within a cluster (when $c = k = \ell$) all links involving i must be ignored:
$$N_{cc}^{+\setminus i} = N_{cc}^+ - r_{ic}^+ - \widehat{r}_{ic}^+ \quad , \quad N_{cc}^- - r_{ic}^- - \widehat{r}_{ic}^-$$

With this approach the sufficient statistics do not need to be recomputed and constantly modified in each Gibbs sweep. This approach provides two main benefits:

- It allows computations of equation 5.1 to be performed in parallel for different nodes i, as the sufficient statistics are no longer temporarily modified depending on the node, that the computations are performed for. To utilize parallel computations, precautions must of course be taken to ensure that the sampling procedure behaves exactly equivalent to the strict serial execution. This will be discussed in section 5.3.

- Computing the beta function now becomes the single most expensive operation when performing the Gibbs sampling (in order to evaluate equation 5.1). To ensure numeric stability the computations are performed in the log-domain such that evaluating the logarithm of the Beta function becomes the key operation (Paper B). To speed up these computations, the model class utilizes an auxiliary data structure that implements lookup tables of precomputed values. This is presented in section 5.2.3.

The model class is close coupled to both the data structures that represent the network A and clustering Z. To efficiently facilitate the Gibbs sampling, the data structure for the clustering must allow for inexpensive looking up cluster assignments and reassigning nodes. The data structure for the network must store links such that the vectors $r_i^+, r_i^-, \widehat{r_i^+}$ and $\widehat{r_i^-}$ can be efficiently computed for any node i. This constitutes efficiently iterating over all links *from* and *to* any node.

5.2 Data structures

The runtime performance of data structures is influenced by multiple factors: How often a new instance of the data structure is constructed, in what way the data is accessed, and how often the stored data is modified. Based on these factors different types of data structures are presented:

- **Constructed and modified seldom**, exemplified by the data structure for assignment matrices. Once the network data is loaded into the data structure it will not be modified during the inference procedure. The main concern for performance is hence the cost of looking up links for a particular node. Memory usage is also a concern for large networks.
- **Constructed seldom and modified often**. This includes data structures that represent the model parameters or sufficient statistics, where the data is expected to change during the inference procedure. In particular we look at clustering of nodes. During the inference this implementation must allow us to efficiently reassign nodes to other clusters, look up the cluster assignment of a given node and iterate over all nodes in a cluster.
- **Constructed and modified often**. These are data structures that have a very short lifespan. They are used to store intermediate computations within functions or subroutines and can often simply be implemented by standard library collections such as vectors or lists. As part of the clustering implementation we use a custom implemented container (partialVector)

to identify which nodes are assigned to a given cluster. This container allows to easily reassign nodes between clusters. It will be presented as an example of a data structure that is both often created (whenever a new cluster is created) and modified (when nodes are reassigned during the inference procedure).

5.2.1 Network data

The examined networks of brain connectivity that were modelled in the included work were represented by undirected adjacency matrices. These were either binary matrices for single subjects or weighted integer matrices for aggregated populations. A generic data structure must however allow for the representation of various types of topologies.

Figure 5.5 shows examples of three types of simple networks; an undirected unipartite network, a directed and weighted unipartite network, and a bipartite network. A network can be represented by a graph $G = (V, E)$, where V denotes the set of vertices (nodes) and E the set of pairwise interactions (edges) between the nodes.

5.2.1.1 Graph topology

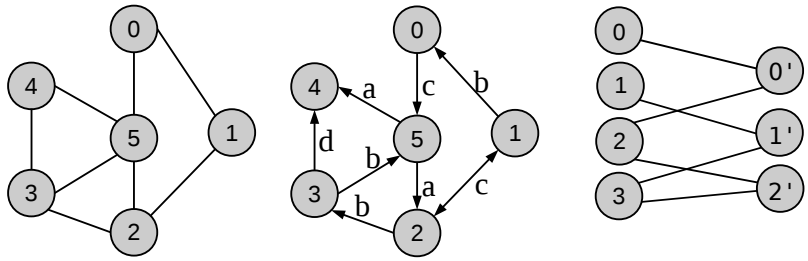
Letting V represent each vertex by a unique integer in a gap-free sequence, each edge in E can be considered a two element set that contains the two integers representing the connected vertices. For the undirected network in figure 5.5a we get:

$$V = \{0, ..., 5\}$$

$$E = \{\{0,1\}, \{0,5\}, \{1,2\}, \{2,3\}, \{2,5\}, \{3,4\}, \{3,5\}, \{4,5\}\}$$

In a directed graph the edges are oriented and can no longer be represented by an unordered set of the two connected vertices. Instead each edge in E can be considered a two-element tuple $[source, target]$, describing the connection from one vertex (the source of the edge) to another vertex (the target of the edge). For the directed and weighted graph in figure 5.5b, each edge is associated with a weight that represents some property of the link between the vertices. An edge can be defined as a three-tuple $[source, target, weight]$ representing the source vertex, target vertex and weight of the edge. For the example we get:

$$E = \{[0,5,c], [1,0,b], [1,2,c], [2,1,c], [2,3,b], [3,4,d], [3,5,b], [5,2,a], [5,4,a]\}$$



(a) Undirected **(b)** Directed, weighted **(c)** Bipartite

Figure 5.5: *Graph representation for simple networks.*

Though numbering the vertices has no other meaning than to identify the individual vertices, it allows for various structured, abstract representations. For the undirected graph in figure 5.5a, figure 5.6 shows the two standard ways to represent a graph.

$$\begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 & 1 & 0 \end{bmatrix}$$

$$\begin{aligned} &⓪ : \{1, 5\} \\ &① : \{0, 2\} \\ &② : \{1, 3, 5\} \\ &③ : \{2, 4, 5\} \\ &④ : \{3, 5\} \\ &⑤ : \{0, 2, 3, 4\} \end{aligned}$$

(a) Adjacency matrix **(b)** Adjacency list

Figure 5.6: *Representations for a simple binary network with $N = 6$ nodes.*

The *adjacency matrix* for a unipartite network is a $|V| \times |V|$ matrix. Each row and column represents a single vertex, such that the weight of a given edge $[source, target, weight]$ is stored at the element $(source, target)$ in the matrix. The time complexity for querying whether a link exist between two nodes is constant $O(1)$, while iterating through all links for a node is asymptotic upper bounded by the number of nodes $O(|V|)$.

The *adjacency list* representation consists of $|V|$ lists, one for each node i in the network. The list for node i represents the adjacent nodes, that are linked to from node i. The cost of querying whether a link exists from node i to node j depends on the size of the list for i. Using binary search the time complexity becomes $O(log(|E_i|))$, where $|E_i|$ is the length of the adjacency list for node i. The complexity for iterating all links for a node is $O(|E_i|)$.

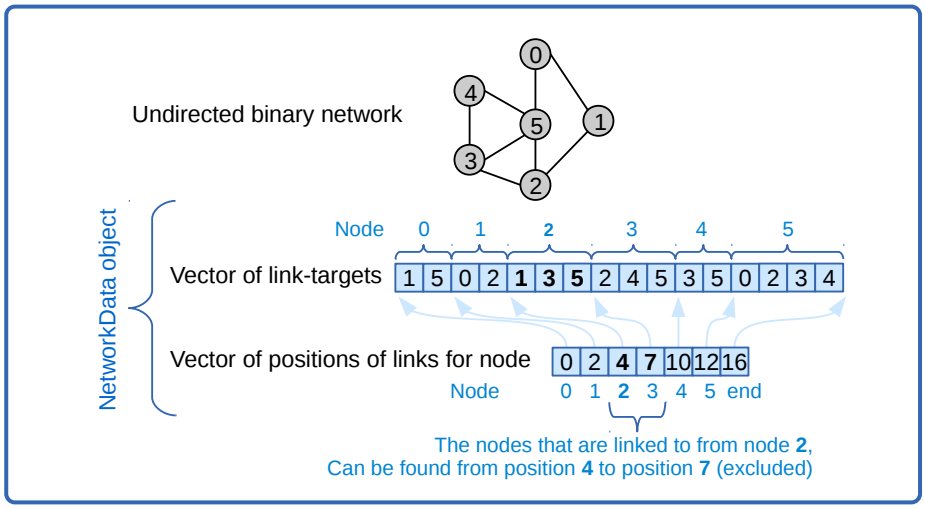


Figure 5.7: Data structure for undirected binary network. *A NetworkData object represents links in the network by two sequential vectors. The first vector stores identifications for all link targets, while the second maps between a given node i and the range within the first vector that contains links with i as source.*

5.2.1.2 Graph implementation

For very small or very dense networks it might make sense to use the assignment matrix representation, which more intuitively resembles the mathematical notation used when describing the statistical model. Further it easier allows the data structure to map directly to memory and allows for the implementation to easier rely on dedicated libraries and hardware for matrix arithmetic and manipulation.

In practice for large sparse networks it holds that $|E| >> |V|$ while $|E_i| << |V|$ for all nodes i, and the preferred representation is often to use adjacency lists [Cormen et al., 2001]. Iterating through links hence becomes significantly cheaper using adjacency lists than using an adjacency matrix. Furthermore the memory required is bounded by $\Theta(|V| \times |E|)$ which is hence also significantly less than $\Theta(|V|^2)$ if storing the assignment matrix. During the MCMC sampling, the key operation involving the network data structure is to iterate through all links for particular nodes. This is done in order to update the sufficient statistics and compute the link-counts between a given node and all clusters.

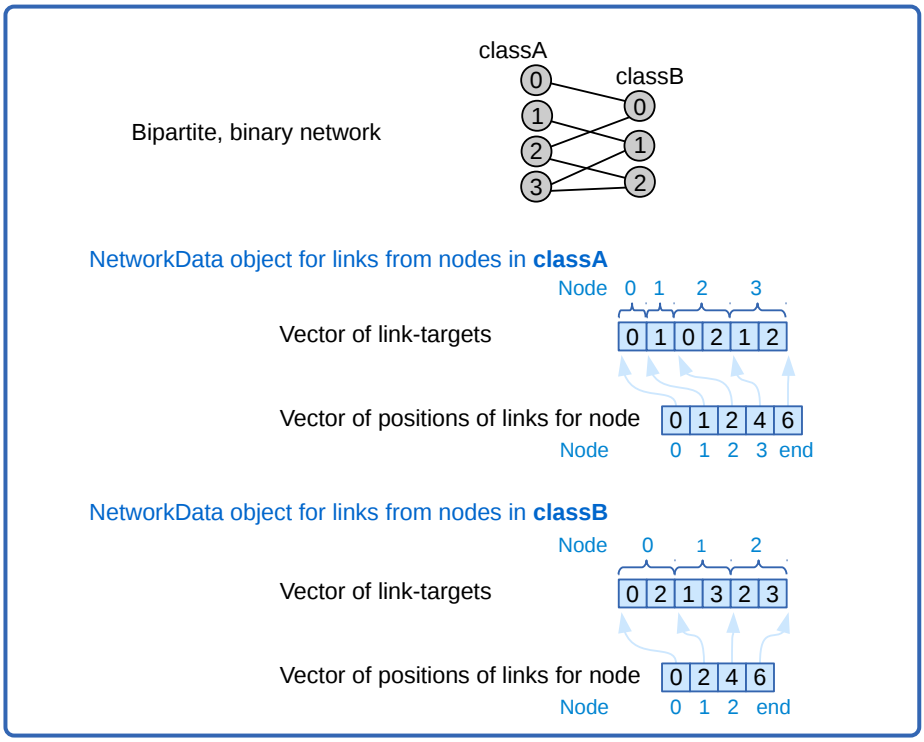


Figure 5.8: Data structure for binary bipartite network. *Two NetworkData objects are utilized to respectively represent links for nodes in the two classes of nodes.*

Figure 5.7 illustrates the data structure that is implemented to contain network data, as instances of the implemented NetworkData class. The figure illustrates the representation for an undirected binary network. This representation can rely on a single NetworkData object, that contain two sequential containers. All adjacency lists are stored sequentially in one vector object. The second vector is used to indicate the range of elements within the first vector that defines the adjacency lists for the individual nodes.

Depending on the network topology multiple NetworkData objects are utilized to store the data and allowing linear time complexity for iterating over links. Figure 5.8 illustrates how two NetworkData objects can be utilized to represent a bipartite network, such that the nodes are separated between the two classes while still allowing linear time complexity $O(|E_i|)$for iterating over all links for any node i.

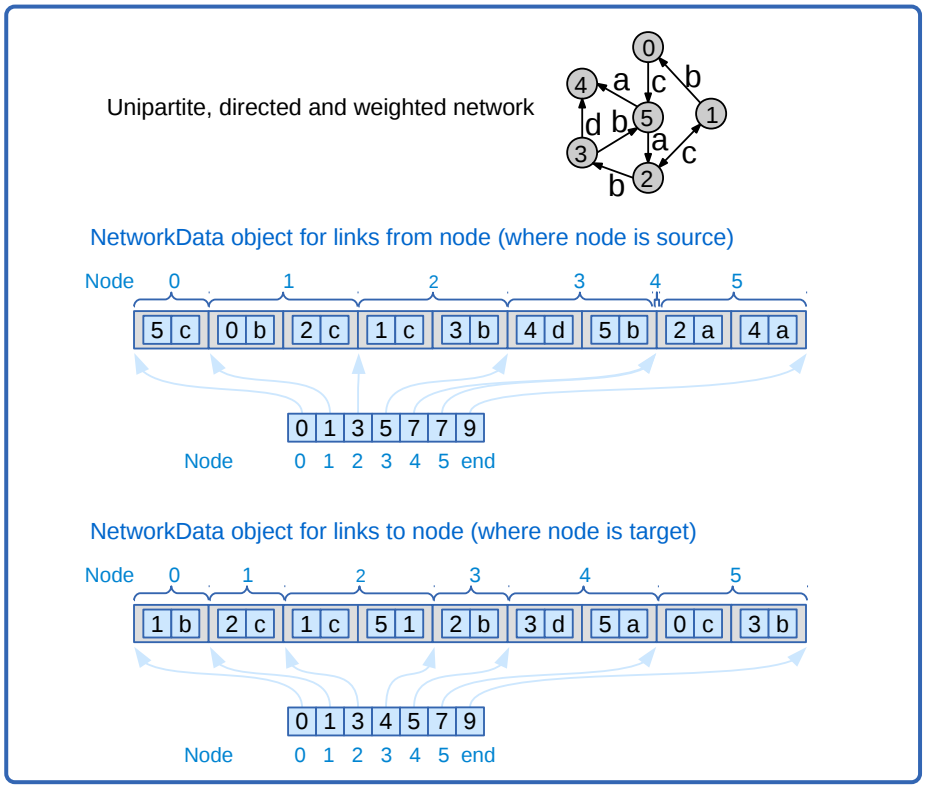


Figure 5.9: Data structure for directed and weighted network. *Two NetworkData objects are used to store links, respectively according to the links target and source nodes.This representation allows for linear time complexity for iterating all links that has either a given node as target or source.*

Figure 5.9 illustrates how NetworkData objects can be used to represent a directed and weighted network. The NetworkData class is implemented such that it can be templated by the type T that represent the weights of the network. Each element in the vector representing the adjacency lists is a two-element struct, with an integer member field representing the node, that is either the target or source for the link (depending on the use of the NetworkData object) and a member field of type T used to represent the weight of the link. To allow linear time complexity for iterating through all links that either has a given node i as target or source, two NetworkData objects are utilized. Binary networks can be implemented as template specializations as they do not need to store information for the weight of links. Here simply the existing of a link-struct defines

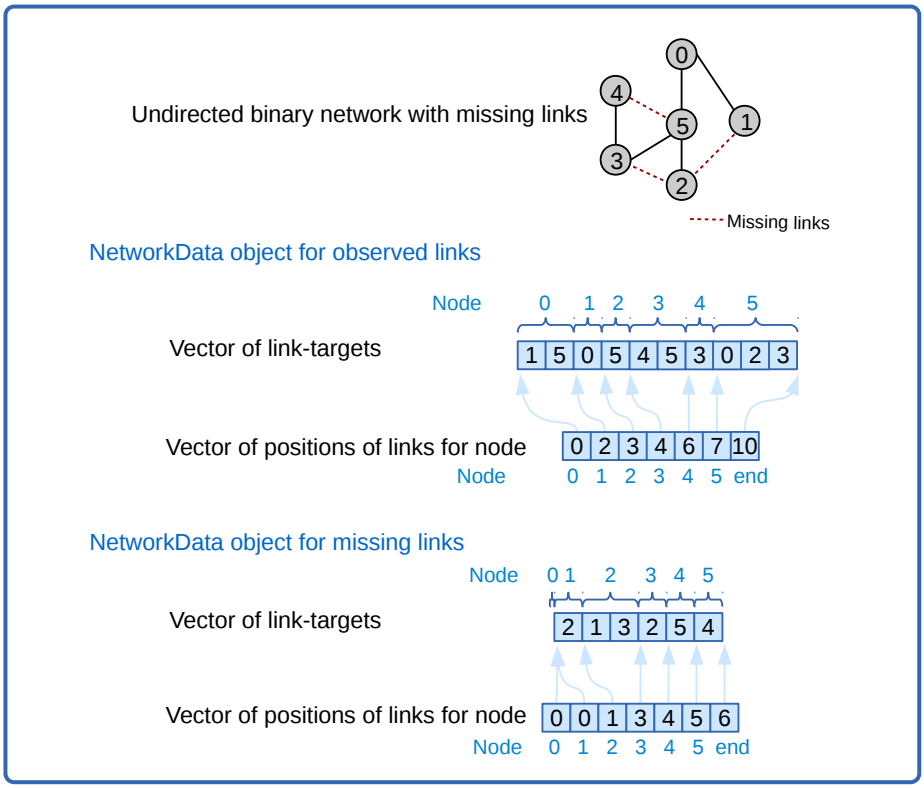


Figure 5.10: Data structure for binary network with missing links. *Links that are treated as missing are represented in a separate NetworkData object (or two objects in the case of directed links).*

that a link exists.

When modelling missing links, dedicated NetworkData objects are used to represent these, as illustrated in figure 5.10 for a binary undirected network.

5.2.2 Clustering data

During the MCMC sampling the clustering parameter $\boldsymbol{z}$ is constantly modified as nodes are reassigned to different clusters. The key operations and the ideal time complexity for accessing and modifying the clustering data are:

- Looking up the cluster assignment of a given node. This constitutes a simple lookup that maps the integer value representing the node to the integer value representing the cluster. It should be performed in constant time $O(1)$.
- Iterating through all nodes in a particular cluster. For any cluster c this operation should ideally be performed in linear time $O(N_c)$ where N_c is the number of nodes assigned to cluster c. Furthermore the data should be stored sequentially in memory to limit random memory lookups and avoid cache misses.
- Reassigning a node to another cluster. This operation involves removing the node from one cluster and inserting it in another. Ideally it must be performed in constant time $O(1)$ and should not modify the data structures such that other operations become more expensive.

Figure 5.11 illustrates the concept for efficient representation of clustering data, by presenting an example of the clustering of a small network. The representation consists of two data structures. The clusterArray-structure is a sequence container where the ℓ'th element represents the particular cluster z_ℓ. Each element ℓ is itself a sequence container of integer values that represent an unordered set of all nodes assigned to z_ℓ. The nodeArray-structure is a sequence container where the i'th element describes the cluster assignment of node i by a struct with two integer members. The first member describes what cluster $\ell = z_i$ the node is assigned to, while the second member describes the position where the node is represented within the container associated with cluster z_i in clusterArray.

By simply reading the i'th element in nodeArray, the cluster assignment of a given node i can be obtained in constant time $O(1)$. Through clusterArray all N_c nodes assigned to a given cluster c can be iterated through in linear time $O(N_c)$.

Reassigning a node i from a cluster ℓ to another cluster m constitutes changes to both containers in clusterArray associated with ℓ and m as well as to the element in nodeArray associated with i. When inserting i in m, an element can simply be added to the container representing m and position i in nodeArray can be updated to reflect the correct cluster number and position within the container now representing node i. Figure 5.12 shows an example illustrating three ways to ensure data consistency when reassigning a node. The position representing node i within a cluster container can be found in constant time as a simple lookup in nodeArray. The relevant computations for removing a node from cluster ℓ depends on the implementation of the cluster container. The default behaviour if using the standard library vector (which is the standard

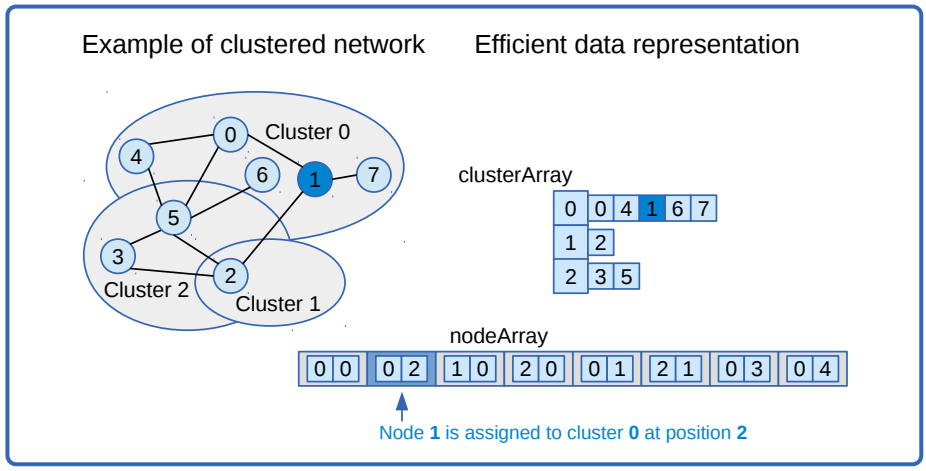


Figure 5.11: Concept for efficient clustering data structure. *Example of data representation for the clustering of a small network with seven nodes, partitioned into three clusters. The data representation relies on two data structures; the* clusterArray *allows efficiently identifying all nodes in a given cluster, while the* nodeArray *allows for efficiently identifying the cluster a given node belongs to and where it is placed in the associated cluster-array.*

C++ container for storing data sequentially) is that all elements after the erased one will be relocated to keep the stored data strictly sequential in memory. The time complexity of this operation is in worst case linear in the number of nodes in the cluster N_ℓ and in worst case necessitates N_ℓ changes to elements in nodeArray. Though the copy operation involved in relocating sequential data in a vector is fairly inexpensive, the affected elements in nodeArray are not necessarily accessed in sequence. A more efficient way of keeping data strictly sequential is to only relocate the last element to occupy the empty position when removing a node. This operation can be performed in constant time.

If the demand for keeping data strictly sequential is relaxed, an auxiliary container can be utilized to keep track of unused positions in the cluster container. Using this auxiliary structure the unused positions can be ignored while iterating the cluster and used to represent the nodes that afterwards are assigned to the cluster. Though this solutions require additional bookkeeping it provides two advantages: *First*, it decouples the implementation of clusterArray and nodeArray. *Second*, it allows the nodes within a cluster to be identified by their unique position for the entire time they are assigned to the cluster.

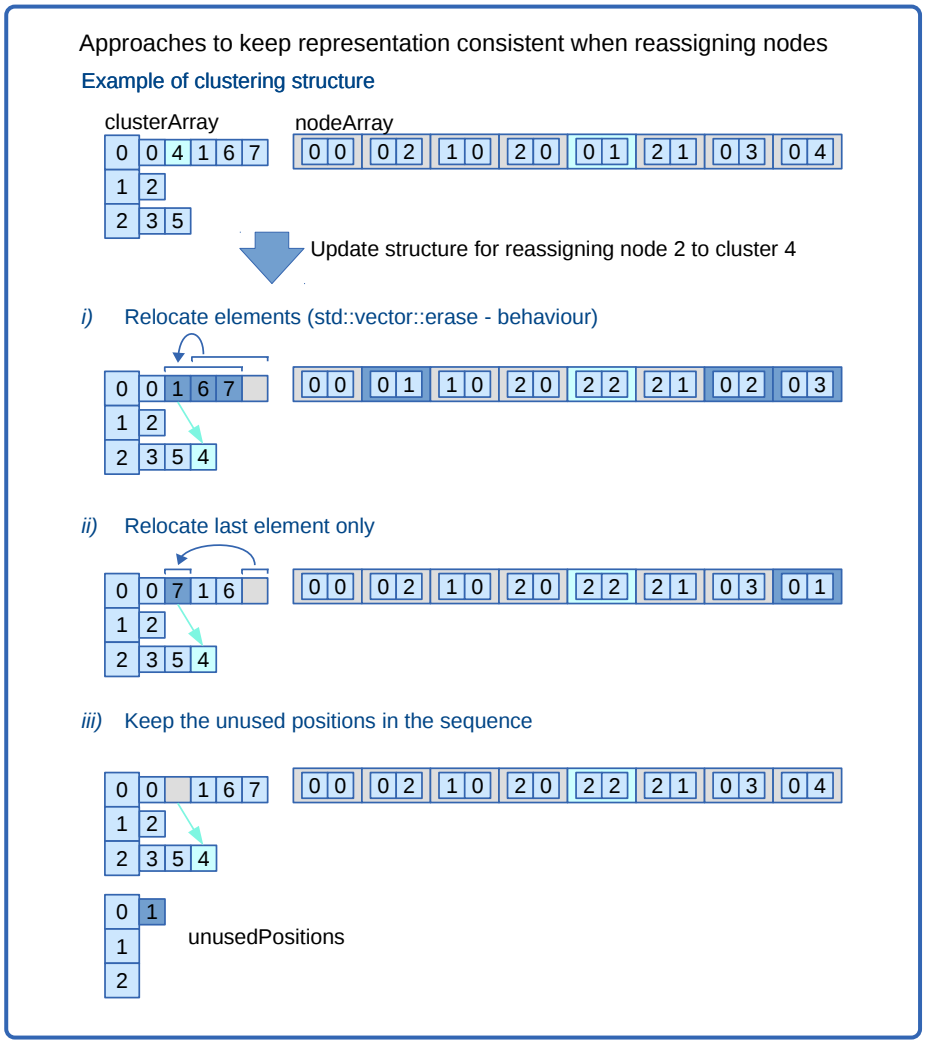


Figure 5.12: Reassigning nodes in the clustering representation. *The figure illustrates three ways to ensure data integrity when reassigning a node. In the shown example, node 4 is moved from cluster 0 to cluster 2. This constitutes changes to the container for cluster 0. i) The container can either be contracted, ii) the last element can be relocated to the now empty position or iii) a separate structure can be used to keep tracks of empty positions in the cluster containers.*

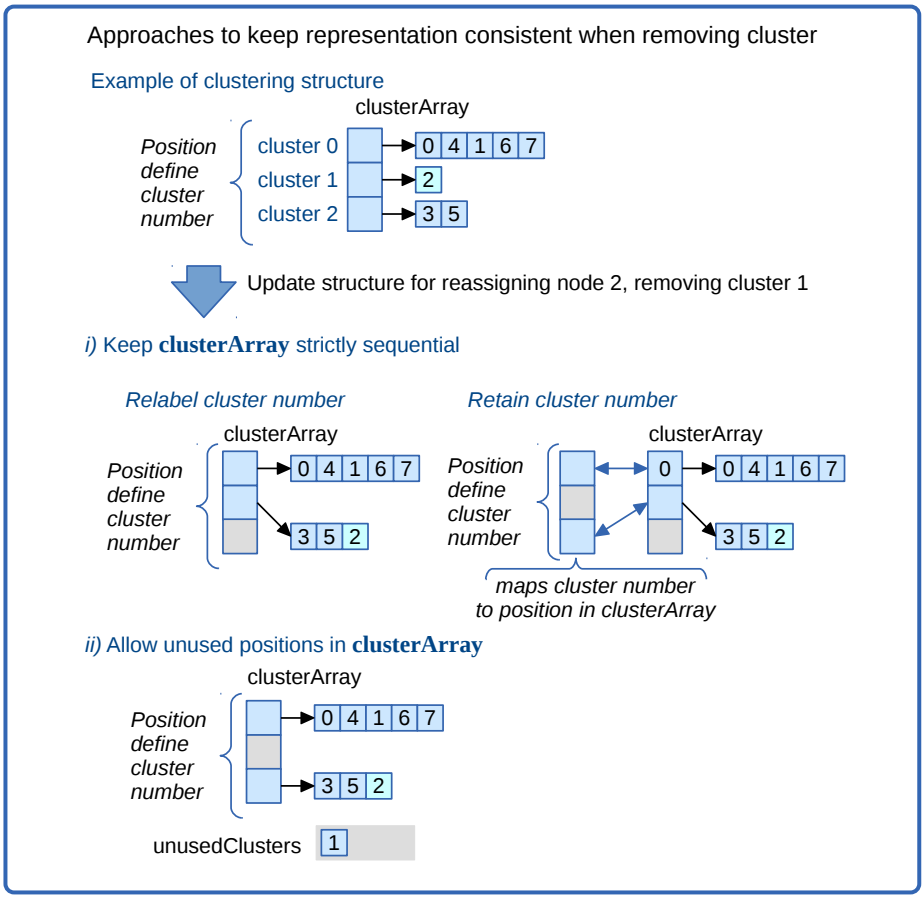


Figure 5.13: Removing empty clusters in the clustering representation. *In the shown example, cluster 1 becomes empty as the only node is reassigned to cluster2. In the Infinite Relational Model the empty cluster is hence no longer part of the clustering and should be excluded, which constitutes changes in the* clusterArray. *To keep* clusterArray *strictly sequential the last cluster can be relabelled to position 1 and a lookup table can be used to map between cluster numbers and positions. Alternatively holes can be allowed in* clusterArray *and a separate structure used to record these, such that they can be assigned newly formed clusters.*

The advantage of such behaviour becomes apparent when considering the implementation for the clusterArray container itself and the problem of removing a

cluster when it becomes empty during inference in the Infinite Relational Model (IRM). This is illustrated in figure 5.13.

The clustering structure for IRM cannot contain empty clusters. clusterArray can be implemented to store pointers to the cluster containers. When a cluster ℓ becomes empty it can simply be removed by swapping the pointer for ℓ with the pointer of the last cluster m. This will however constitute N_m changes in nodeArray to record that all nodes in cluster m is now in cluster ℓ. To avoid this, an auxiliary structure can be utilized to map between the cluster number and its position within clusterArray. Alternatively clusterArray can be allowed to contain empty positions, using an auxiliary structure to record these positions such that they can be ignored while iterating all clusters and used when new clusters are created.

From the statistical modelling point of view the number associated with a cluster is irrelevant. The practical implementation however relies on sufficient statistics recording the number of links and non-links between clusters, which can only be intuitively decoupled from the clustering implementation if cluster numbers are never changed. A similar argument can be made for keeping the position of individual nodes within the cluster containers constant. Though the toolbox currently does not contain any models that rely on this (such as hierarchical clusterings), the generic implementation allows for such utilizations. Further more allowing empty positions in both clusterArray and the cluster containers allows the implementation to just rely on the single custom-implemented container type called a partialVector.

5.2.2.1 partialVector

The partialVector is a collection data-structure that provides the benefits of both a sequential array and a linked list. It keeps the data elements stored sequentially in memory such that it is fast to iterate through while it preserves constant time insert, push and delete operations. The data-structure is implemented as a class template and can hence be generically utilized.

Figure 5.14 illustrates how an element is removed and added to a partialVector. The partialVector is implemented as an object utilizing two stl sequence containers: a vector and a deque (double-ended queue). The vector contains the actual data elements, though it is allowed to contain holes (being unused positions). The deque is used to keep track of these unused positions in the vector, such that new elements can be added to an empty position in the vector in constant time. When a new element is pushed to the partial vector, it will be inserted in the vector in an unused location found by popping the deque. If

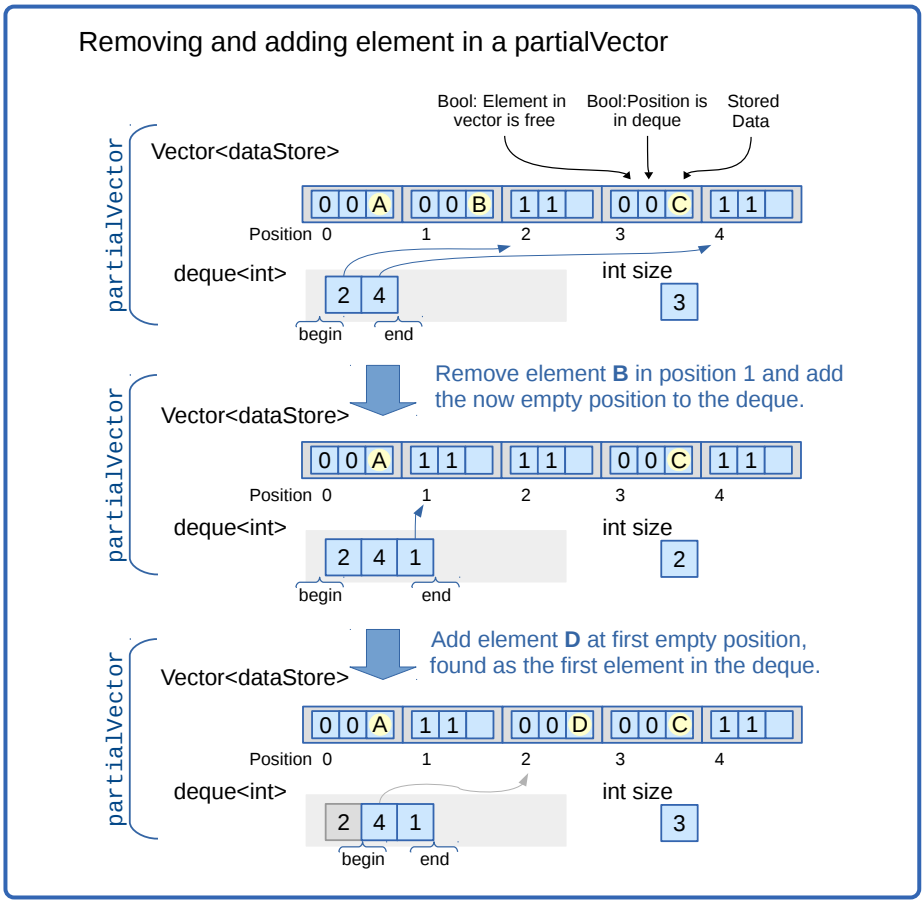


Figure 5.14: Removing and adding elements using a partialVector. *In this example, initially the partialVector contains three elements A,B and C. These are stored in the vector at position 0,1 and 3. The vector hence contains two empty positions 2 and 4, which are remembered in the deque. When the element at position 1 is removed, a value is added to the deque showing that the position is now not in use. When an element is added, it is inserted at a random empty position, found as the first element in the deque.*

there are no unused positions, the newly inserted element will be pushed to the back of the vector. The push function will return the position within the vector that the element is stored in. By providing this position, the user can recover the stored data. Access to random elements (including insert, push and delete)

can now be done in constant time. The expected behaviour of the partialVector deviates from that of a normal vector object as deleting/erasing an element does not causes the collection to relocate all elements after the erased one, which can be an expensive operation for larger collections.

Besides the actual data element, each element in the vector also contains two bookkeeping Boolean values, that are not exposed outside the partialVector object. The first describes whether the position in the vector is unused and hence should be ignored while iterating. The second Boolean describes whether the position in the vector can be found in the deque of empty positions. The information is stored in a struct dataStore with three members being the two Boolean values and the actual data. As shown in figure 5.14) the value type stored in the vector are hence instances of such structures. The partialVector implements an iterator class that implements the std forward-iterator interface. The iterator traverses the dataStore structs by implementing an iterator for the vector object. Dereferencing the partialVector::Iterator pointer returns the data member of the referenced dataStore struct, while the increment operator (++) re-references the iterator to the next position that is not empty according to the bookkeeping Booleans. When iterating over the elements in the partialVector, holes will hence be ignored. With balanced number of insert/push and delete operations, the expected iteration time will be linear in the number of elements. As the elements are stored in sequential memory, iterating will be faster than using a linked list.

The bookkeeping further allows that an element can both be added to a random empty position (as in figure 5.14) or inserted at a particular position defined by the user. This functionality makes the data structure more generic and has been utilized to load predefined and randomly generated clusterings into the application (Paper C). As bookkeeping the partialVector also keeps count of the number of non-empty elements in the vector. This is necessary as the vector object can contain holes and its length hence not always represent the size of the partialVector.

In the implemented toolbox we rely on partialVector-objects to represent both the individual clusters, the clusterArray and the nodeArray within the clustering data structure. It is further utilized as an auxiliary data-structure throughout the application.

5.2.3 Lookup tables

In a previous studies we have examined how the runtime of MCMC sampling in IRM depends on using different approximations for the logarithm to the gamma

function (which will be denoted $\mathsf{gammaln}(x)$) to compute the logarithm of the beta function:

$$log(\mathsf{Beta}(a,b)) = \mathsf{gammaln}(a) + \mathsf{gammaln}(b) - \mathsf{gammaln}(a+b). \tag{5.2}$$

We found that compared to the C++ library function, the MCMC sampling could only be performed twice as fast before deviations of more rough approximations significantly influenced the result of the inference (Paper B). To obtain a more significant speedup we utilize a lookup table of precomputed gammaln values. In Paper D we did not sample the hyper-parameters that were fixed at $\beta = \beta^+ = \beta^- = 1$. In this case it is trivial to use a lookup table of precomputed values. Such a table can be implemented simply as an array where the i'th element stores the value gammaln(i).
When β is not an integer value, the lookup table can simply store evaluations of gammaln that includes β, such that the i'th element stores the value gammaln$(i+\beta)$, as the counts for links and non-links are integer values.
To allow for different values for the two hyper parameters we can resolve to using three tables; respectively computed when the evaluation depends on (β^+), (β^-) and $(\beta^+ + \beta^-)$. The i'th element in the three tables respectively stores the value gammaln$(i+\beta^+)$, gammaln$(i+\beta^-)$ and gammaln$(i+\beta^+ + \beta^-)$.

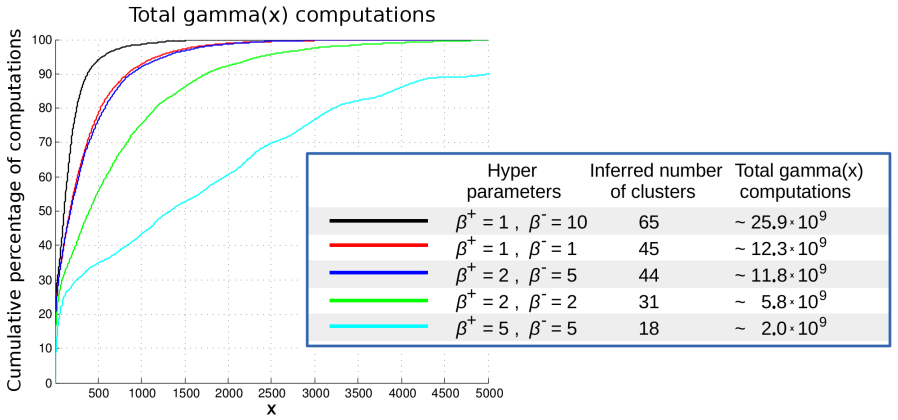


Figure 5.15: *Total number of gamma(x) computations during IRM for 1000 Gibbs sweeps on the Hagman network with 998 nodes. For all experiments the concentration parameter is* $\alpha = 10$.

Figure 5.15 shows the total number of calls to the gammaln(x)-function during 1000 Gibbs sweeps for IRM on the Hagman network for various β^+ and β^- values. The figure indicates that most gammaln(x) computations are for lower values of x. For instance when $\beta^+ = \beta^- = 1$ the figure indicates that a lookup table with $2,500$ elements can cover more than 95% of all function calls, such

that less that 5% must be computed by more expensive approximations.

To allow hyper-parameter sampling, we simply use function calls for approximations of gammaln(x) during the hyper-parameter sampling, where β^+ and β^- are expected to change, and recompute the entire lookup tables afterwards. As the sequence of gammaln values can be iteratively computed;

$$log(\Gamma(x+1)) = log(\Gamma(x)) + log(x) \quad \text{for} \quad x > 0,$$

the cost of recomputing a lookup-table depends linearly on the cost of evaluating the logarithm function.

For larger networks (such as for the modelled HCP data in Paper E) the cost of recomputing a lookup-table is affordable. Here it can take hours to complete a single Gibbs sweep and the lookup-table is rarely recomputed. For smaller networks the Gibbs sampling is so fast, that it becomes a concern to find a reasonable trade off between the benefit of using a larger sized table to the cost of recomputing the table.

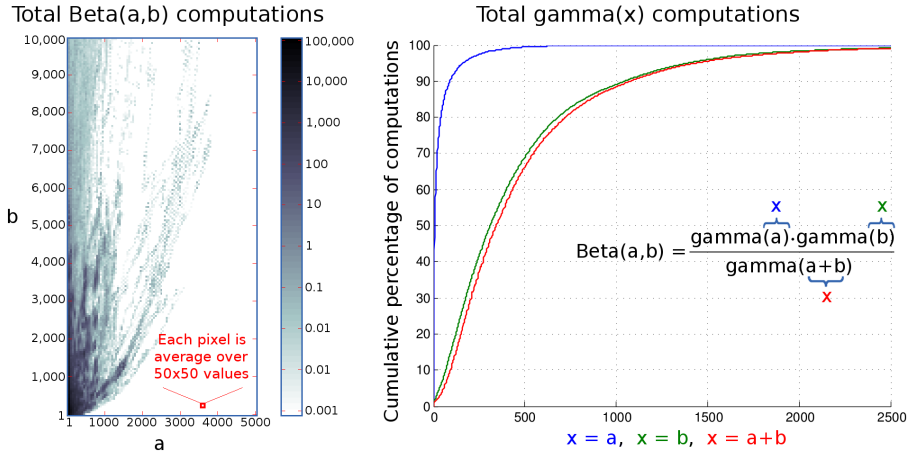


Figure 5.16: *Hagman network, Gibbs sampling alone, $\beta^+ = \beta^- = 1, \alpha = 10$. The shows the averaged results for 5 restarts, each with 1000 Gibbs sweeps.*

Figure 5.16 separates between gammaln()-calls involving β^+, β^- and $\beta^+ + \beta^-$, such as when using three lookup tables. As expected the figure indicates that the lookup table involving only β^+ can be much smaller than the other tables. The figure also illustrates the distribution of the total log-Beta computations. Alternatively to using three gammaln tables (which results in three random fetches from main memory for each log-Beta lookup), we can use a single two

dimensional lookup table of precomputed log-Beta values, which will only result in a single random fetch for each lookup. The following table shows the percentage of all log-Beta lookups that are covered by various sizes of such a table for the data in figure 5.16:

stored values	columns (|a|)	rows (|b|)	coverage of Betaln(a,b) requests
5,000	50	100	12.5%
10,000	50	200	27%
100,000	100	1,000	81%
1,000,000	100	10,000	91%

As expected the table shows that significantly more values must be stored to obtain the same coverage when using a lookup table of log-Beta values than when using three tables of log-gamma values. For large problems it hence seems unfavourable to use this approach, even though it constitutes fewer random memory lookups.

5.3 Parallelization

Reconsider the pseudo-code in Figure 5.2 for performing a single Gibbs sweep. The code contains two loops: The outer loop iterates over every node i, while the inner loop iterates over every cluster k in order to obtain the posterior change of assigning node i to cluster k.

We have experimented with parallelizing both loops, using OpenMP to express thread-level parallelism. OpenMP provides a set of compiler directives and library routines that can extend C++ to easily implement shared-memory parallelism [Dagum and Menon, 1998]. This is done by inserting various compiler pragmas in the code, If the directive pragmas are ignored the parallelized code can easily be compiled as sequential code and executed without any runtime overhead.

5.3.1 Parallelization within Gibbs iteration

The inner loop of the Gibbs sweep can intuitively be parallelized by letting each of T threads be responsible for computing $\frac{K}{T}$ iterations of the loop.

To avoid excessive virtual function calls, section 5.1.1 presented how the inner loop was replaced with a single function call, that instead received a vector (as

argument) representing the clusters to be considered. By moving the implementation of the inner loop into a function implemented by the Model object, the K virtual function calls were replaced by just a single virtual call.
In the parallel version, each of the T threads simply calls this function with a vector containing only the $\frac{K}{T}$ clusters associated with the thread. A barrier ensures that values for all threads are obtained after which a single thread is responsible for evaluating what cluster the node must be assigned to.

When only parallelizing the inner loop the Gibbs sweep, some computations are still left to be evaluated sequentially by a single thread. This constitutes the computations for obtaining the categorical distribution of assignment probabilities and computing the new cluster assigning for the considered node.

5.3.2 Parallelization over Gibbs iterations

In order to parllelize the outer loop of the Gibbs sweep algorithm, the cluster assignment for multiple nodes must be evaluated in parallel. Shared with MCMC methods in general, the Gibbs sampling procedure is inherently sequential. If node i is reassigned the probability distribution for cluster assignments for all the next nodes are likely to be altered. The cluster assignment for node i and $i + 1$ can in that case not be computed in parallel when we wish to keep the sampling procedure behaving equivalent to strict serial execution.

Instead we can utilize the parallel resources using various procedures for *predictive prefetching* [Byrd et al., 2010, Strid, 2010, Angelino et al., 2014], such that cluster assignment for node i and $i + 1$ are computed in parallel under the assumption that node i will be assigned to a particular cluster. Only if it later turns out that this assumption was correct, will the computed cluster assignment for node $i + 1$ be applied. Otherwise it must be recomputed. We utilize an approach similar to *speculative moves* presented by Byrd et al. [Byrd et al., 2008], with the assumption being that nodes will *not* be reassigned, which in practise seems to be a likely assumption for longer chains of Gibbs sampling.

Our procedure is illustrated in Figure 5.17 using three threads, but generalizes to T threads. The cluster assignment for node $i + 1$ is computed under the assumption that node i is not reassigned, in which case the probability distribution for assigning node $i + 1$ is not altered. When the cluster assignment for node i is finally obtained, we know whether the assumption was correct. If node i was in fact not reassigned, the parallel computed cluster assignment for $i + 1$ is correct and can be applied. If node i was assigned to another cluster, the computed cluster assignment for node $i + 1$ is invalid and must be recomputed using the correct cluster assignment for node i.

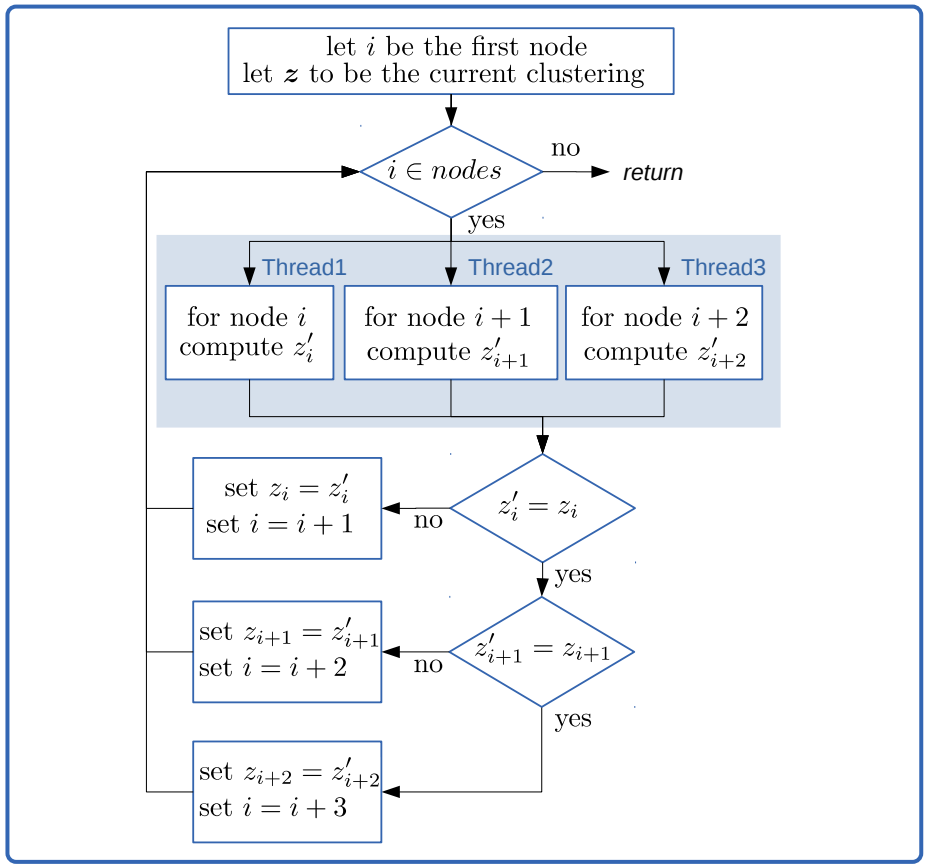


Figure 5.17: Diagram for computing a Gibbs sweep, *illustrating how multiple threads can be used to parallelize the outer loop of the algorithm for computing a single Gibbs sweep. In each loop in the diagram, three threads are used to speculatively compute the cluster assignment for three nodes in parallel. This is done under the assumption that cluster assignments for nodes with lower numbers (that are considered in parallel) are not changed. Only a single node can hence be reassigned in each iteration.*

The more cores that are utilized the more nodes will be evaluated in parallel, which makes it more likely that some node will be reassigned and more computational resources will be wasted. Figure 5.18 illustrates this, for Gibbs sampling performed on the averaged network of brain connectivity with 998 nodes [Hagmann et al., 2008]. For different number of cores, the figure presents

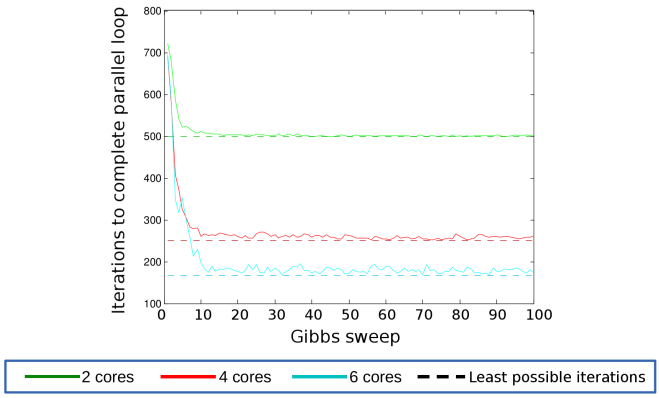


Figure 5.18: Iterations of outer parallel loop. *Iterations of the parallel loop to iterate over all 998 nodes when doing Gibbs sampling in IRM on the averaged network of brain connectivity with 998 nodes [Hagmann et al., 2008]. The figure shows results for a single chain using 2, 4 and 6 cores and marks the least possible iterations (which would be the case if nodes were never reassigned). The sampling procedure were performed for 100 sweeps, only consisting of only Gibbs sampling with fixed hyper-parameters* $\beta^+ = \beta^- = 1, \alpha = 10$..

the number of iterations of the parallel loop that was performed in order to traverse all 998 nodes in each Gibbs sweep.

traversed in order to complete the Gibbs sweep.

5.4 Computational speedup

The runtime performance of Gibbs sampling in IRM is shown in Figure 5.19 for the averaged network of brain connectivity with 998 nodes [Hagmann et al., 2008]. The figure compares the two ways of parallelizing the Gibbs sampler as well as the impact of using the table lookups for computing the gammaln() function.

The figure shows that using table lookups allows the sampling procedure to run almost 10 times as fast for a single thread. In a practical setting we find that multiple cores are likely to be better spend by running multiple chains of the sampling procedure in parallel in order to assess convergence, compare predictive performances and evaluate different sampling strategies. In situations

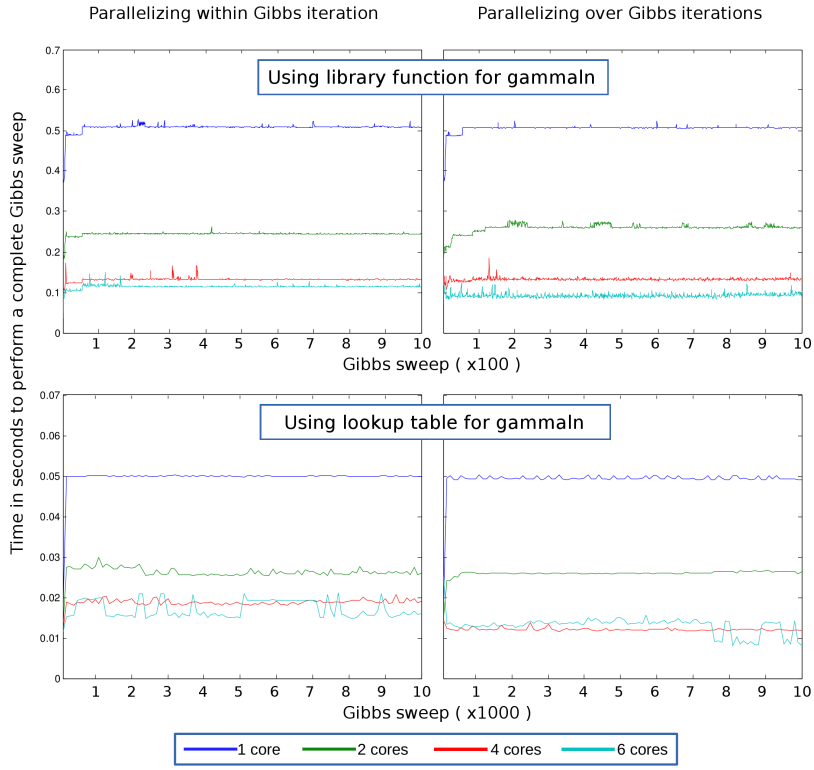


Figure 5.19: Speed up for Gibbs sampling using multiple cores. *Hagman network with 998 nodes.* $\beta^+ = \beta^- = 1, \alpha = 10$. *The figure shows the results for 1000 sweeps using the library function for gammaln()-computations and 10,000 sweeps using table lookups.*

where multiple chains cannot be executed in parallel do to a tight computational memory budget, the chains must be executed sequentially. In such situations parallelizing the sampling procedure of a single chain is a valid approach to decrease the total computation time. In this test it seems like both approaches to parallelization yields similar speedups, with almost linear speedup when using two or four threads. In this test it seems that there is little advantage in using 6 threads instead of 4. This can be caused by hardware limitations (such as memory bandwidth) or the particular problem size and highlights the importance of testing different configurations for a particular problem in order to optimally utilize the resources.

Chapter 6

Research contributions

The included papers utilizes the implemented software tools (at different stages of development and in some cases slightly modified versions) for two different purposes. In data analysis projects the software solutions provide practical means for obtaining clusterings of whole-brain connectivity based on large scaled networks obtained from dMRI and fMRI data. Furthermore the generic designs allows the software to be easily modified and is utilized for exploring various computational and statistical properties of the stochastic blockmodels when applied to different sized networks.

6.1 Paper A: The Influence of Hyper-Parameters in the Infinite Relational Model

The infinite relational model with Bernoulli likelihood relies on the Beta prior for modelling the independent, identically distributed link probabilities. In this paper we investigate the influence of different hyper-parameter configuration for the Beta prior and the influence of sampling these parameters.

On three real world networks of different sizes, we investigate three different prior constructions: Joint symmetric ($\beta^+ = \beta^-$), joint asymmetric (as presented in section 3.2.3) and separate asymmetric (where different pairs of parameters are used for within and between clusters). We sample the hyper-parameters using the Metropolis-Hastings procedure (section 3.3.3). These results of sampling the parameters are also compared with using fixed values for the hyper-parameters that are either uninformed or based on the link density of the network.

We find that the hyper-parameter configuration significantly influences both the number of inferred clusters and the predictive performance of IRM. Compared to inference based on a symmetric prior configuration, sampling with an asymmetric prior configuration allowed the model to identify a more refined block-structure and showed a better predictive performance on hold out data. Sampling with separate parameters for within and between clusters showed an even better predictive performance and a more refined block-structure without indicating over fitting to the training data.
When sampling the hyper-parameters in the asymmetric prior configuration, the model performed on par with using fixed values based on the network density if scaled appropriately. This scale factor could not be intuitively chosen. We found that inferring the hyper-parameters not only reflect the link density of the entire network, but reflects the average link densities for the identified blocks. Appropriate values for the hyper-parameters can not be guessed in advance (based only on the observed network) but can be identified by the sampling procedure.

6.2 Paper B: Numerical Approximations for Speeding Up MCMC Inference in the Infinite Relational Model

In an efficient implementation of Gibbs sampling in the infinite relational model with Bernoulli likelihood and Beta prior, the computationally demanding op-

eration becomes evaluating the logarithm of the Beta function which relies on evaluating the logarithm of the Gamma function. In this paper we investigate the influence of using different numerical approximations for computing the log-Gamma function. We evaluate the influence both in terms of the compute time for performing the inference procedure and in terms of how the inferred clustering is affected by the numerical precision.

The influence of different numerical approximations is evaluated and compared by performing IRM on the same test data for all approximations, and further compared to using machine precision by utilizing the standard lgamma() function in C++ (declared in the C numerics library).

We find that introducing numerical approximations influences the complexity of the inferred clusterings. For instance, Stirling's approximation which is very imprecise when evaluated for small values introduces significant bias into the model, such that the MCMC procedure converges to solutions with lower test likelihood while distinctively identifying more clusters. Notably, the Lanczos 1.5 approximation provides results similar to using the standard library function but computes twice as fast. From the experiments we can conclude that using approximations can speed up the inference procedure, but not to an extend that significantly improves the runtime or compares to using lookup tables of precomputed values (as presented in section 5.2.3).

6.3 Paper C: Local Modes in the Posterior Distribution of Dirichlet Process Mixture Models

When modelling larger networks, the MCMC sampling procedure cannot reach convergence within reasonable time. This is particularly an issue for procedures like the Gibbs sampler, where escaping local modes often involves reassigning multiple nodes as reassigning the nodes individually does not yield a posterior gain. In this paper we characterize the local modes of the posterior distribution for the Infinite Relational Model with Bernoulli likelihood, in order to investigate the influence of the sampling procedure getting "trapped" in such local solutions.

Local solutions are obtained by using the hill-climbing optimization procedure presented in section 3.3. Experiments are performed on synthetic networks and smaller real world networks of various complexity, in order to investigate:

- The number of local solutions and how they differ and compare to the global solution.

- The “basins of attraction” of local modes for evaluating how easy the MCMC sampler might get stuck and what the penalty of inferring local solutions is, compared to solutions found by the converged MCMC sampling.

The number of local solutions identified in real world networks appears to mimic the number of solutions in the random graphs with similar number of edges. Compared to the number of nodes, the real world networks have substantially fewer local solutions that the random graphs, likely to be caused by the examined networks being more sparse.

By comparison the clusterings of the most often found local solutions with the global solutions with highest posterior value we outline how the sampling procedure might "escape" being stuck in the local solutions. For smaller networks a few split-merge operations seems to be enough to move from the most often found to the best solution. For larger network it seems like more advanced operations would be beneficial, as up to nearly half of the nodes needs to be repositioned. For the real world networks we however see that often found local solutions tends to be fairly good and for some networks even approaches the posterior of the global solution.

Based on these studies it seems that even though the sampling procedure cannot converge and might get stuck in local solutions, the inferred clusterings can still to a fair extend account for the connectivity structure in the network, which is also what we can conclude from clustering large complex networks in the other included papers. However, it seems that more advanced sampling operations are necessary to avoid the sampling procedure getting trapped in local modes.

6.4 Paper D: Nonparametric Bayesian Clustering of Structural Whole Brain Connectivity in Full Image Resolution

In this paper we investigate whether it is feasible to use a data-driven approach for extracting the latent clustering information for whole brain connectivity based on diffusion MRI in full image resolution. After preprocessing and binarization the connectivity networks were obtained from raw dMRI scans and contained in the order of one hundred thousand regions and one hundred million links.

The Infinite Relational Model (IRM) was utilized to extract the latent clustering

structure inferred by using our MCMC sampling procedure without sampling the hyper-parameters. Though the MCMC procedure could not converge we find that the model could be computationally scaled to handle data of this size and complexity and infer clusterings that describe some of the latent structure in the data.

Quantified by normalized mutual information we find that repeated runs of IRM infer similar clusterings, which also to a great extend is in agreement with two existing anatomical atlases; the Destrieux atlas [Fischl et al., 2004, Destrieux et al., 2010] and the Desikan-Killiany atlas [Desikan et al., 2006].
As a non-parametric model, IRM can by itself identify an appropriate number of clusters based on the data. While the two anatomical atlases respectively separates the connectome into 148 and 68 regions, IRM identifies approximately 1000 clusters. The clusterings inferred by IRM further shows a better predictive performance than the atlases, when predicting a connectivity network obtained from a rescan of the same subject. Even though atlases can be subdivided to obtain a finer resolution [Hagmann et al., 2008] the division will not be based on the structural connectivity. The results show that IRM is capable of capturing much of the subject specific structure from data. By not using a fixed number of structural units, IRM can more freely fit the clustering to the complexity of data. However, as discussed in section 3.6, this can also cause IRM to overfit to the training data if the goal is to predict connectivity networks obtained from other subjects.

6.5 Paper E: Predictive Validation of Human Brain Parcellation

Analysis of brain connectivity is often based on various anatomical atlases. In this paper we address the problem of quantifying the quality of such different brain parcellations, by a presented predictive framework where the quality of a parcellation is evaluated by statistical prediction on independent brain connectivity data.
We use independent high-resolution data of structural brain connectivity obtained from the Human Connectome Project (HCP) to compare the predictive performance of the atlases.
We test three particular atlases: Desikan-Killiany [Desikan et al., 2006], Destrieux [Fischl et al., 2004, Destrieux et al., 2010] and a HCP provided atlas [Glasser et al., 2016]. While the two former atlases are based solely on surface morphology and has a low resolution of respectively 68 and 148 regions, the later combines multiple modalities and has a much higher resolution of 360

regions. Furthermore we compare the predictive performance of the atlases with parcellations obtained through various data-driven approaches - including finite Stochastic Block Modelling with MCMC sampling as presented in chapter 3.

We find that all atlases perform significantly better than what would be expected from random parcellations. The multi-modal HCP atlas is clearly better at accounting for the structural connectivity data. It even performs on par with the SBM which is trained and optimized to describe the structural data. The data-driven approaches however reveal that the structural connectivity data is better accounted for when using more regions than the three atlases have Furthermore we see that parcellations derived from larger populations significantly improves the predictive performance. The larger the population size, the more regions can be accounted for to obtain the optimal predictive performance. We however see that a broad range of resolutions result in an almost optimal predictive performance.

6.6 Paper F: Joint Modelling of Functional and Structural Whole-Brain Connectivity

The brain can be studied by its different modalities, such as by its structural and functional connectivity. The brain can be assumed to fundamentally be organized into computational units that acts as precursors from which the observable structural and functional connectivity emerges. If this assumption is true then integrating both modalities when inferring a data-driven parcellation might aid in recovering connectivity for the individual modalities. The extend to which structure and function is related and how to quantify this relation remains challenging.

In this paper we use the stochastic block model to jointly infer clusterings based on both modalities. As quantified by AUC when predicting test networks of each modality, we compare the predictive performance of multi-modal integration with clustering inferred from the modalities individually.
We use data from the HCP database (as in Paper E), separated into groups of different population sizes of 1, 2, 10 and 50 subject. For each group an averaged network of functional and structural connectivity is obtained. The networks are binarized and threshold to 1 percent density. The predictive performance are evaluated on similarly created hold-out networks with 50 subjects for both modalities. For all experiments we limit SBM to 360 clusters, in order to more fairly compare the results with using the multi-modal HCP provided atlas [Glasser et al., 2016] that contains 360 parcels.

By using averaged networks for different population sizes we effectively manipulate the signal-to-noise ratio of the data. For predicting both modalities the AUC score improves for larger population sizes while the standard deviation decreases. When predicting functional connectivity we find that multi-model integration significantly improves the performance for all population sizes, while it hampers the performance when predicting structural connectivity. In Paper E we demonstrated that the atlas is highly qualified for predicting structural connectivity. This finding is confirmed as the atlas is capable of predicting both the structural and functional test data with a high AUC. The predictive performance for the atlas also improves for larger population sizes but the model based approaches presents a more pronounced improvement. AUC is always higher when predicting structural data also when comparing the raw graphs. This likely reflects that the dynamic functional data is more prone to noise - which also might explain why we observe that multi-modal integration aids in predicting functional but not structural connectivity.

6.7 Paper H.1: Predictive Evaluation of Human Value Segmentations

This paper is not included in the thesis, but serves to illustrate that the methodology and implemented tools intuitively can be expanded to slightly different models and be applied in different research areas that relies on different data sources. The following short synopsis is included for the interested reader, likewise is the full paper printed as supplementary information in appendix H.1.

Within social sciences, survey studies provide data for quantifying human value priorities, with various segmentation methods providing means for characterizing response patterns within the survey data. Evaluating the quality of segmentations however remains challenging. This paper illustrates how Bayesian mixture modelling can be used to obtain sound segmentations and provide predictive evaluations for the quality and comparison of segmentations.

The paper compares segmentations of human values survey data from the forth round of European Social Study (ESS-4), and shows that demographic markers such as age or nationality predicts better than random but is significantly outperformed by the Bayesian approach. Respondents are clustered based on their binarized responses to the question items, using the Bayesian mixture model

defined by the generative process:

$$
\begin{aligned}
z_i &\sim \mathit{Dirichlet\text{-}Categorical}(\alpha) && \text{Clustering of respondent } i \\
\eta_{lq} &\sim \mathit{Beta}(\beta_q^+, \beta_q^-) && \text{Response probability} \\
A_{iq} &\sim \mathit{Bernoulli}(\eta_{z_i q}) && i \text{ answers 'yes' to question } q
\end{aligned}
$$

The clusterings are inferred using a combination of full and restricted Gibbs sampling while the hyper-parameters are inferred using the Metropolis-Hastings procedure presented in section 3.3.

The study shows that social studies can benefit from using generative probabilistic modelling, which provides statistical salient means for evaluating the model fit and comparing different segmentations. The paper illustrates that human value priorities transcend geographical boundaries and divides humans into more complex personality types, as the inferred segmentations reflects the expected trend that value priorities are shared across geographical regions while some local and national characteristics are still present.

Chapter 7

Discussion and conclusion

This project was motivated by the aim of modelling networks of whole-brain functional and structural connectivity in the high resolution supported by modern MRI techniques. Based on experience obtained from implementing and using the developed stochastic block modelling tools, we limit our discussion to the three related topics:

- How to computationally scale the model implementation to handle complex networks of the size obtained from high-resolution MRI.
- How the implemented models and inference procedure behave and compare when applied for large-scale modelling.
- What insight to the organization of the human brain we have obtained and can further expect from data-driven parcellations of whole-brain connectivity.

We have demonstrated that it is possible to scale the models to computationally handle whole-brain connectivity networks in very high resolution. To obtain the necessary computational performance we found it essential to highly optimize the implementation by taking both hardware and model properties as well as runtime usage into consideration.
The most significant speedup of the MCMC procedure we actually obtained by

identifying and managing the special function that are called most frequent. Function calls to special functions such as the logarithm and gamma function are individually fast, but the functions are called so intensively that significant speedup is obtained through either faster approximations or through table lookups of precomputed values if the model distributions allow for this.
MCMC algorithms are conceptually sequential and the expected speedup of parallel execution is hence limited, depending on the particular algorithm and data. One should however not refrain from using parallel resources even when faced with a sequential problem. We find that extensive optimization might leave the most time consuming work actually being accessing data and table lookups, which can benefit from parallel fetches.

Though the dedicated high-performance implementation allows the MCMC procedure to be executed extremely fast, it never converges to the target distribution. We find that the posterior space for all network sizes contains an astounding number of local modes that the samplers tend to get drawn towards and stuck in. Substantial efforts are put into designing MCMC procedures that converge faster by being more manoeuvrable in the high-dimensional spaces [Behrens, 2008], having shorter burn-in periods and benefitting naturally from parallel computations [Mahani and Sharabiani, 2015]. However, for the problem sizes we investigate, MCMC procedures that reliably can explore the entire posterior space are not imminently available.
As the procedure never converges one might think that it negates the entire purpose of using MCMC sampling, and simply relax the problem to stochastic optimization. In many situations, we however observe that a random sample provides significant better predictions than simply optimizing towards a local posterior maximum. This shows that even though the sampling procedure does not converge or express the full potential of MCMC, the procedure can still successfully be applied to large scale modelling and the inferred parcellations will capture the underlying structure in the data.
A further important model aspect in capturing network structure is the hyper-parameter configuration. We find it essential for the examined class of models that the hyper-parameters are learned. When examine both real-world and synthetic data we find that the hyper-parameters have a huge impact for aiding in learning and recovering the network structure. We further find that they cannot intuitively be set in advance, but can straightforward be inferred as part of the MCMC procedure.

The Infinite Relational Model (IRM) tends to identify many smaller and singleton clusters. When optimizing towards a local posterior mode, the tendency is for such small clusters to be aggregated or absorbed into larger clusters. We have observed how this tendency can actually hurt the predictive performance on hold-out data. The cluster size distribution is hence not simply a spuriosity of the model. It seems that IRM actually identifies and utilizes small clusters

to more precisely describe connectivity properties of the networks.

On real-world brain connectivity data, the non-parametric nature of IRM also provides parcellations that contains many small clusters. Based on large networks of structural whole-brain connectivity (Paper D and Figure 3.12) we have seen that IRM identifies in the order of thousands of clusters for a single subject. This allows IRM to have a high predictive performance for rescans of the same subject. The non-parametric approach however also have two major drawbacks. *First*, the many small clusters can make the inferred parcellation difficult to interpret and compare with other parcellations, such as anatomical atlases that only contain in the range of a few hundred parcels. *Second*, we observe that in some situations the non-parametric clustering prior gives too much freedom to the model, in particular when coupled with the freedom of sampling the hyperparameters. When the goal is to predict the population average based on limited data, the parcellations from IRM might fit too closely to the training data (section 3.6), resulting in a sub-optimal predictive performance. The covariate shift between training and test data therefore presents IRM as a slight model to data mismatch in these cases.
The model must hence be regularized towards the population somehow. This might have been achieved by using fixed hyper-parameters or introducing some population prior. We have however chosen to tune the model by directly manipulating the number of clusters, by using the parametric stochastic blockmodel (SBM). If the goal had been to solely obtain optimal predictions on other subjects, this might not have been the ideal solution. SBM however contains some very nice properties: *First*, it is a very simple model and provides solutions that are easy to interpret. *Second*, SBM has a fixed number of clusters, which allows us to easily compare parcellations of different sizes and directly compare with the fixed sized anatomical atlases.
Having a fixed number of clusters is of course also a drawback of SBM when we wish to identify an appropriate optimal number of clusters. In Paper E we do this by cross-validation which can be computationally intensive.

Using the implemented tools we have demonstrated that stochastic block modelling successfully can be applied for clustering whole-brain connectivity. We have shown that brain connectivity can be meaningfully partitioned by the purely data-driven approach. The un-supervised models can learn spatial homogeneous regions without being informed of spatial information at all. The identified spatial regions are meaningful; they are in agreement with existing atlases and allow for good predictions on hold-out data.
The stochastic block model is a generative model of data. In Paper E we present how the model also intuitively can be used as a validation tool to quantify the quality of parcellations. This is done by statistically assessing how well the parcellations characterizes connectivity structure as quantified by predictions on independent test data. The more data the model is trained on, the closer

the training data will reflect the population, which allows for better predictive performance. When training on population sized networks, we find that there furthermore is statistical support for identifying a more complex connectivity structure, as evident by SBM being capable of partitioning the data into more clusters for an optimal predictive performance. Whether the increased complexity actually describe real organization in the brain or are driven by systematic biases in the training and test data remains an open question.
Using the predictive framework, we can evaluate the quality of anatomical atlases and the data-driven parcellations obtained from SBM. We compare with random parcellations based on k-means, which forms spatial homogenous clusters. We find that both atlases and model inferred solutions all predicts significantly better than using random parcellations. This suggests that the data-driven approach of SBM seems to be in compliance with the anatomical atlases in capturing the organization of the brain.

Particular strong in describing the structural connectivity was the newly released multi-modal HCP_ MMP1.0 atlas [Glasser et al., 2016]. For limited data, this atlas almost predicted on par with our model, which was trained specifically on similar data. The HCP_ MMP1.0 atlas is based on multiple modalities including task and resting-state fMRI, but not directly on the structural dMRI we used for predictions. The convincing capability of this atlas to describe structural connectivity is a compelling argument for the idea that the structural organization of the brain is interdependent with the other modalities.
In Paper F we investigate this hypothesis. Here we use SBM to obtain parcellations from structural and functional data individually as well as by jointly modelling both modalities and comparing the inferred parcellations with the HCP_ MMP1.0 atlas.
Structural and functional connectivity describes two very different properties of the brain and presents very different connectivity profiles.
However both modalities share information, which to some extend is expected if they both emerges from the same fundamental organization from the brain. The functional data obtained by fMRI is very noisy. We see that integrating structural information improves the predictions of functional connectivity. By using data from both modalities, the data-driven approach can learn a functional parcellation of the brain, that is more in compliance with functional data than a parcellation inferred from functional data alone. By training SBM on multiple subjects, we can tune the signal-to-noise ratio of the fMRI data. With a large population of 50 subjects, we still observe that joint modelling predicts better than using functional data alone.

I find the future perspective for the presented modelling framework very promising. As a practical tool it allows us to examine the organization of large scaled complex networks, integrate data from different sources, and predictively compare and evaluate results. This approach can intuitively be utilized in multiple

domains beyond modelling brain connectivity. In the future I expect to see even more complex data, higher resolutions and increased use of data-driven modelling in general. The presented approach for scaling Bayesian relational modelling supports a beneficial data-driven approach, that might be capable of handling the amount and complexity of data expected in the future.

Appendix A

The Influence of Hyper-parameters in the Infinite Relational Model

The Influence of Hyper-parameters in the Infinite Relational Model.
Kristoffer J. Albers, Morten Mørup, and Mikkel N. Schmidt. In Machine Learning for Signal Processing, IEEE International Workshop on, (MLSP), 2016.

2016 IEEE INTERNATIONAL WORKSHOP ON MACHINE LEARNING FOR SIGNAL PROCESSING, SEPT. 13–16, 2016, SALERNO, ITALY

THE INFLUENCE OF HYPER-PARAMETERS IN THE INFINITE RELATIONAL MODEL

Kristoffer J. Albers, Morten Mørup, Mikkel N. Schmidt

Department of applied Mathematics and Computer Science, Section for Cognitive Science
Technical University of Denmark

ABSTRACT

The infinite relational model (IRM) is a Bayesian nonparametric stochastic block model; a generative model for random networks parameterized for unipartite undirected networks by a partition of the node set and symmetric matrix of inter-partion link probabilities. The prior for the node clusters is the Chinese restaurant process, and the link probabilities are, in the most simple setting, modeled as iid. with a common symmetric Beta prior. More advanced priors such as separate asymmetric Beta priors for links within and between clusters have also been proposed. In this paper we investigate the importance of these priors for discovering latent clusters and for predicting links. We compare fixed symmetric priors and fixed asymmetric priors based on the empirical distribution of links with a Bayesian hierarchical approach where the parameters of the priors are inferred from data. On synthetic data, we show that the hierarchical Bayesian approach can infer the prior distributions used to generate the data. On real network data we demonstrate that using asymmetric priors significantly improves predictive performance and heavily influences the number of extracted partitions.

Index Terms— Infinite relational model, hyper-parameter inference, link-prediction, Bayesian non-parametrics.

1. INTRODUCTION

Many systems, both naturally occurring and engineered, can be described as complex networks. These include biological systems such as functional and structural brain connectivity, social and economic behaviour as well as infrastructure such as power grids, communication and transport networks.

Network science is concerned with developing theoretical and practical methods for modelling and quantifying hidden structure in complex networks, and plays a prominent role in acquisition of knowledge within many different research areas. One way to extract information from a complex network is to cluster the network into groups of nodes that have similar structural connectivity patterns.

The most prominent statistical tool for clustering network data is the stochastic block model [1, 2], which is a probabilistic generative model for random networks. It models a network using a latent clustering of the network nodes. The probabilities of links between two nodes depend only on their cluster assignments and a link probability parameter which is defined for each pair of clusters. In the infinite relational model (IRM) [3, 4] the prior for the cluster structure is the Chinese restaurant process: A stochastic process which defines a distribution over partitions. The CRP provides a nonparametric Bayesian mechanism for learning the number of clusters that best fit the observed network.

The prior for the link probability parameters are, in the most simple setting, chosen as a symmetic Beta distribution. Without any further information available, a vague symmetric prior such as a Beta$(\frac{1}{2}, \frac{1}{2})$ (arcsine) or Beta$(1, 1)$ (uniform) distribution is suited. If more prior information is available, such as beliefs about the overall link density of the network or belief that the link densities within and between clusters are different, using a more elaborate prior is relevant.

In this paper we investigate how different prior constructions in the IRM influence the learned clustering structure as well as the predictive performance of the fitted model. In particular, we demonstrate that using an asymmetric informative prior leads to superior predictive performance compared to other constructions.

2. METHOD

2.1. Review of the infinite relational model

Let A be the adjacency matrix of a simple graph. Including separate parameters for the Beta priors for links within and between clusters, the infinite relational model (IRM) is [3] is given by the following generative

This project was supported by the Lundbeck Foundation, grant nr. R105-9813

978-1-5090-0746-2/16/$31.00 ©2016 IEEE

process:

$$z \sim \text{CRP}(\alpha) \quad \text{Clusters} \quad (1)$$
$$\eta_{\ell\ell} \sim \text{Beta}(\beta_{\text{w}}^{+}, \beta_{\text{w}}^{-}) \quad \text{Link probabilities within} \quad (2)$$
$$\eta_{\ell m} \sim \text{Beta}(\beta_{\text{b}}^{+}, \beta_{\text{b}}^{-}) \quad \text{—} \quad \text{between} \quad (3)$$
$$A_{ij} \sim \text{Bernoulli}(\eta_{z_i,z_j}) \quad \text{Observed links} \quad (4)$$

The prior for the clustering is a Chinese Restaurant Process (CRP), which allows the model to automatically infer an appropriate number of clusters from data. The probability of observing a link between two nodes i and j follows a Bernoulli distribution, where the parameter η_{z_i,z_j} depends on the cluster assignments of the two nodes. In our setup, the link probabilities $\eta_{\ell m}$ within and between clusters follow separate Beta distributions.

We investigate the following different prior constructions: A joint symmetric prior with only one parameter, $\beta = \beta_{\text{w}}^{+} = \beta_{\text{w}}^{-} = \beta_{\text{b}}^{+} = \beta_{\text{b}}^{-}$ as proposed in [3], a joint asymmetric prior with two parameters, $\beta = \{\beta_{\text{w}}^{+} = \beta_{\text{b}}^{+}, \beta_{\text{w}}^{-} = \beta_{\text{b}}^{-}\}$ as used for block modeling in [5], and separate asymmetric priors for link probabilities within and between clusters with four parameters $\beta = \{\beta_{\text{w}}^{+}, \beta_{\text{w}}^{-}, \beta_{\text{b}}^{+}, \beta_{\text{b}}^{-}\}$.

Because the Beta prior is conjugate to the Bernoulli likelihood, the link probabilities ($\eta_{\ell m}$-parameters) can be marginalized analytically, revealing the following joint distribution

$$P(A, z|\alpha, \beta) = \text{CRP}(z|\alpha) \prod_{\ell} \frac{\text{B}(N_{\ell\ell}^{+} + \beta_{\text{w}}^{+}, N_{\ell\ell}^{-} + \beta_{\text{w}}^{-})}{\text{B}(\beta_{\text{w}}^{+}, \beta_{\text{w}}^{-})} \prod_{\ell<m} \frac{\text{B}(N_{\ell m}^{+} + \beta_{\text{b}}^{+}, N_{\ell m}^{-} + \beta_{\text{b}}^{-})}{\text{B}(\beta_{\text{b}}^{+}, \beta_{\text{b}}^{-})}. \quad (5)$$

Here $N_{\ell m}^{+}$ and $N_{\ell m}^{-}$ are the number of links and non-links between cluster ℓ and m, and $\text{B}(a, b) = \frac{\Gamma(a)\Gamma(b)}{\Gamma(a+b)}$ is the Beta function. We can then further place priors on the parameters β in a Bayesian hierarchical manner. In the following we employ improper flat priors, such that the joint distribution can be written as $P(A, z, \beta|\alpha) \propto P(A, z|\alpha, \beta)$.

2.2. Inference using Markov chain Monte Carlo

To solve the clustering problem, we condition on the observed network to find the posterior distribution of the clustering by, $P(z|A, \alpha)$. To infer the clustering we employ two different transition kernels: Gibbs sampling and split-merge sampling. In Gibbs sampling, we loop over each node in the network: For each node we evaluate the posterior distribution when assigning the node to each of the existing clusters or a new empty cluster, conditioned on all the other node assignments $z_{\setminus i}$. The node is then reassigned based on the probability distribution of possible node assignments. The probability of assigning node i to cluster m is then given by:

$$P(z_i = m|A, z_{\setminus i}, \alpha, \beta) = \frac{P(z_i = m, A, z_{\setminus i}|\alpha, \beta)}{\sum_{\ell} P(z_i = \ell, A, z_{\setminus i}|\alpha, \beta)}, \quad (6)$$

where ℓ in the sum ranges over all existing groups and a new empty group.

In split-merge sampling [6], two nodes in the network are selected uniformly at random. If the nodes are in the same cluster it is proposed to be split, otherwise the clusters of the two nodes are proposed to be merged. The proposals are accepted or rejected based on the Metropolis-Hastings acceptance probability:

$$P(z^*|z) = \min\left[1, \frac{P(z^*, A|\alpha, \beta^+, \beta^-) q(z|z^*)}{P(z, A|\alpha, \beta^+, \beta^-) q(z^*|z)}\right], \quad (7)$$

where $q()$ is the transition probability and z^* is the proposed clustering. To generate the proposed split state, the two selected nodes are placed in separate clusters and the remaining nodes in the cluster are allocated randomly between the two. A number of rounds of Gibbs sampling is performed (restricted to the nodes in the two clusters), and the final proposal and its transition probability is then given by the final Gibbs round. For a split configuration, $q()$ is given by the product of the individual transition probabilities of repartitioning each node from the launch state to the split configuration. As there is only one way of merging two clusters, the transition probability for merging clusters is always one.

To infer the parameters of the prior, β, we use a Metropolis-Hastings procedure: We sample each parameter in turn using a Gaussian proposal distribution centered on the current value and with variance $\sigma^2 = 1$.

2.3. Data and experiments

As a generative model, IRM can be used to generate synthetic data. We use this to investigate how well IRM with the different prior configurations is capable of inferring the underlying true parameters and clustering on a synthetic network. We further investigate IRM with the various prior configurations on three real world network data of various sizes and from different domains. These networks are presented in table 1.

We consider the following three prior constructions:

Prior	Parameter(s)	
Joint symmetric [3]	One:	$\beta = \beta_{\text{w}}^{+} = \beta_{\text{w}}^{-} = \beta_{\text{b}}^{+} = \beta_{\text{b}}^{-}$.
Joint asymmetric [5]	Two:	$\beta^{+} = \beta_{\text{w}}^{+} = \beta_{\text{b}}^{+}$, $\beta^{-} = \beta_{\text{w}}^{-} = \beta_{\text{b}}^{-}$.
Separate asymmetric	Four:	β_{w}^{+}, β_{w}^{-}, β_{b}^{+}, β_{b}^{-}.

Network	Nodes J	Links L	Link ratio $a^+ = \frac{L}{P}$	Nonlink ratio $a^- = \frac{P-L}{P}$	Description
USAir	332	2123	0.0387	0.9613	Traffic network of airlines [7], binarized as in [8].
Hagmann	998	37,926	0.0762	0.9238	Average of five brain connectivity networks in [9].
Facebook	4,039	88,234	0.0108	0.9892	Social circles from Facebook [10, 11].

Table 1: Topology for the examined networks. P denotes the total possible links, computed as $P = J(J-1)/2$.

We compare sampling for these constructions using the following fixed symmetric, uninformed priors:
$\beta^+ = \beta^- = \{\ 0.05\ ,\ 0.5\ ,\ 1\ ,\ 5\ \}$,
and using fixed values based on the link- and nonlink-ratio found empirically in the network data (shown in table 1):
$\beta^+ = c \cdot a^+ \quad , \quad \beta^- = c \cdot a^- \ , \quad \text{for } c = \{0.1, 1, 2, 10\}.$
We use the following MCMC sampling procedure for 1000 iterations, where the first 750 iterations are discarded as burn in. Each iteration consists of: One Gibbs sweep over all nodes followed by 10 split-merge proposals, each with three restricted Gibbs sweeps. When sampling the hyper-parameters, 10 proposals for each of the sampled parameters are then performed in each iteration. We consider the concentration parameter α of the CRP fixed at $\log(J)$, where J is the number of nodes in the network. For sampling the β parameters we use a Gaussian distribution with variance 1.

To compare the clustering found by IRM with the ground truth of synthetic data, we use normalized mutual information, $\mathrm{NMI}(z, z') = \frac{2\mathrm{I}(z,z')}{\mathrm{H}(z)+\mathrm{H}(z')}$, where $\mathrm{H}(z)$ is the entropy of the clustering z.

To compare the sampling procedures on real world networks, we evaluate the predictive performance based on the inferred clusterings. When sampling a real world network, we exclude 10 percent of the links as hold out data and measure the predictive performance by evaluating the area under the receiver operating characteristic curve (AUC) when predictions are made for the hold out data [12]. For a given clustering, we compute the expected probability of a link between two clusters as:

$$\langle \eta_{\ell m} \rangle = \frac{N^+_{\ell m} + \beta^+_{\mathrm{b}}}{N^+_{\ell m} + N^-_{\ell m} + \beta^+_{\mathrm{b}} + \beta^-_{\mathrm{b}}} \tag{8}$$

$$\langle \eta_{\ell \ell} \rangle = \frac{N^+_{\ell \ell} + \beta^+_{\mathrm{w}}}{N^+_{\ell \ell} + N^-_{ll} + \beta^+_{\mathrm{w}} + \beta^-_{\mathrm{w}}} \tag{9}$$

The expected probability of a link between two nodes is considered the link probability between the two clusters, the nodes belongs to: $\langle A_{ij} \rangle = \langle \eta_{z_i, z_j} \rangle$. When examining the AUC, we compare averaging over the last 250 iterations of the MCMC sampling and using the estimate for the last iteration only.

For fixed asymmetric priors we use the a^+ and a^- ratio based on the entire network. Instead of modelling the hold out data as missing [13], we treat it as non-existing links in the network [14, 15]. This is a more conservative link prediction strategy that is more prone to overfitting and can hence easier show whether IRM will exhibit overfitting issues when sampling the hyper-parameters.

Hyperparameters	NMI	NOCs
Fixed, symmetric		
$\beta^+ = 0.05\ ,\ \beta^- = 0.05$	0.9502 ± 0.0083	20
$\beta^+ = 0.1\ ,\ \beta^- = 0.1$	0.9789 ± 0.0017	26
$\beta^+ = 0.5\ ,\ \beta^- = 0.5$	0.9502 ± 0.0083	20
$\beta^+ = 1\ ,\ \beta^- = 1$	0.9532 ± 0.0076	20
$\beta^+ = 5\ ,\ \beta^- = 5$	0.9502 ± 0.0083	20
Fixed, empiric		
$\beta^+ = 0.1 \cdot a^+\ ,\ \beta^- = 0.1 \cdot a^-$	0.9903 ± 0.0014	34
$\beta^+ = 1 \cdot a^+\ ,\ \beta^- = 1 \cdot a^-$	0.9913 ± 0.0017	33
$\beta^+ = 2 \cdot a^+\ ,\ \beta^- = 2 \cdot a^-$	0.9875 ± 0.0012	30
$\beta^+ = 10 \cdot a^+\ ,\ \beta^- = 10 \cdot a^-$	0.9652 ± 0.0030	24
Fixed, ground truth		
$\beta^+ = 0.1\ ,\ \beta^- = 1.5$	0.9923 ± 0.0000	35
Inferred		
$\beta = \beta^+_{\mathrm{w}} = \beta^-_{\mathrm{w}} = \beta^+_{\mathrm{b}} = \beta^-_{\mathrm{b}}$	0.9778 ± 0.0016	28
$\beta^+ = \beta^+_{\mathrm{w}} = \beta^+_{\mathrm{b}}, \beta^- = \beta^-_{\mathrm{w}} = \beta^-_{\mathrm{b}}$	0.9921 ± 0.0005	35
$\beta^+_{\mathrm{w}}, \beta^-_{\mathrm{w}}, \beta^+_{\mathrm{b}}, \beta^-_{\mathrm{b}}$	0.9919 ± 0.0005	35

Table 2: Normalized Mutual Information (NMI) and number of components (NOCs) found by sampling IRM on a synthetic network with $J = 500$ nodes and 17.324 links, generated from an IRM with $\alpha = log(J)$, $\beta^+ = \beta^+_{\mathrm{w}} = \beta^+_{\mathrm{b}} = 0.1$ and $\beta^- = \beta^-_{\mathrm{w}} = \beta^-_{\mathrm{b}} = 1.5$. The true clustering contains 35 components. The results are based on five random restarts each averaged over the last 250 iterations of the sampling procedure.

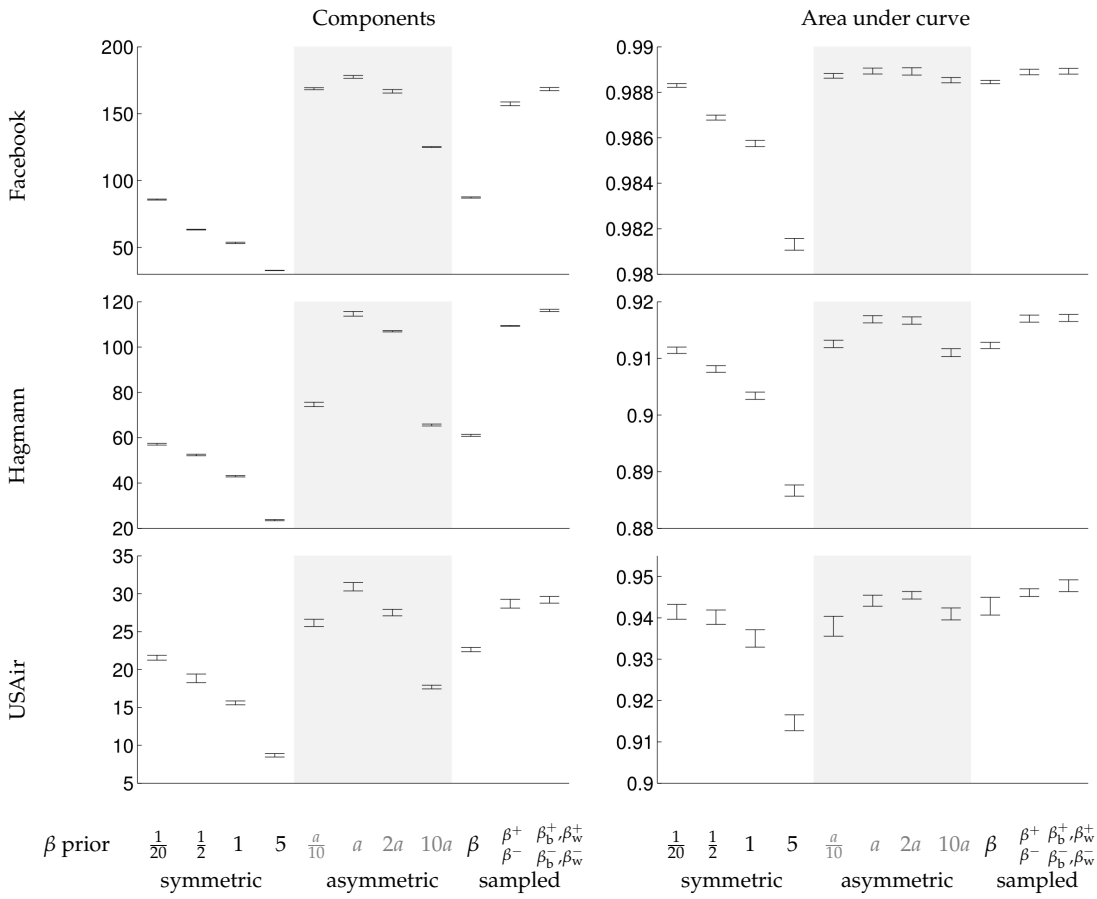


Fig. 1: Number of components and AUC for the real world networks for the different prior constructions on β. The sampling procedures were performed with 1000 sweeps for five restarts on five different hold out set for each network. The number of components was averaged over the last 250 sweep. AUC is computed from the averaged clustering over the last 250 sweeps. Errorbars indicate the standard deviation of the mean over five restarts, i.e. $\text{std}/\sqrt{5}$.

Network	Density	Joint asymmetric		Separate asymmetric			
		Mean cluster density	$\rho = \frac{\beta^+}{\beta^+ + \beta^-}$	Mean cluster density within	Mean cluster density between	$\rho_w = \frac{\beta^+_w}{\beta^+_w + \beta^-_w}$	$\rho_b = \frac{\beta^+_b}{\beta^+_b + \beta^-_b}$
USAir	0.0348	0.2100	0.1787	0.4556	0.1676	0.3819	0.1451
Hagmann	0.0686	0.0788	0.0780	0.8336	0.0689	0.8200	0.0674
Facebook	0.0097	0.0253	0.0231	0.4968	0.0202	0.4639	0.0197

Table 3: The inferred values of the hyperparameters compared to mean cluster densities of the inferred clustering and the density of the training network. Results are averaged for the last 100 sweeps of the sampling procedures for a single run.

3. RESULTS AND DISCUSSION

3.1. Synthetic data

Table 2 shows the average normalized mutual information as well as average number of inferred components when sampling with the different prior constructions in a synthetic network generated from an IRM. The network is generated with a joint assymetric prior for $\beta^+ = 0.1$, $\beta^- = 1.5$ and the generated clustering contains 35 components. When using a fixed symmetric prior the model under-estimates the number of components, while using fixed asymmetric priors allows the model to better adapt to the network. The model can correctly identify values for β that corresponds to a high NMI if the hyperparameters are fixed appropriately.

Sampling two or four parameters both correctly identify the number of components. The inferred clusterings further seem to have a similar NMI, just as good as found when using the ground truth hyperparameter values for the generated network. Sampling the hyperparameters, on average gives the following values for the three hyperparameter settings:

A: β	$= 0.106 \pm 0.003$
B: β^+	$= 0.096 \pm 0.003$, $\beta^- = 1.628 \pm 0.043$
C: β_b^+	$= 0.097 \pm 0.004$, $\beta_\mathrm{b}^- = 1.718 \pm 0.076$
β_w^+	$= 0.132 \pm 0.005$, $\beta_\mathrm{w}^- = 1.437 \pm 0.100$.

Thus, hyperparameter values fairly close to those used to generate the network are identified when sampling two or four parameters.

3.2. Real world networks

Figure 1 shows the results with the various prior configurations on the three real world networks. Ten percent of the networks were omitted as holdout data for computing the AUC. Supporting the findings from sampling on synthetic data, the model performs better when based on asymmetric priors. It identifies more components with a higher AUC. While using fixed asymmetric priors based on network topology can perform well it requires an appropriate scale of the parameters is chosen, which might depend on the particular network. When sampling the asymmetric hyper parameters separately between and within components, the model is capable of identifying more components retaining the same high AUC as sampling the parameters as joint asymmetric.

Figure 2 shows the accumulative sizes of the inferred clusterings when sampling with the different prior configurations. The additional clusters found when using asymmetric priors are not dominated by small or singleton clusters, suggesting that they contain relevant

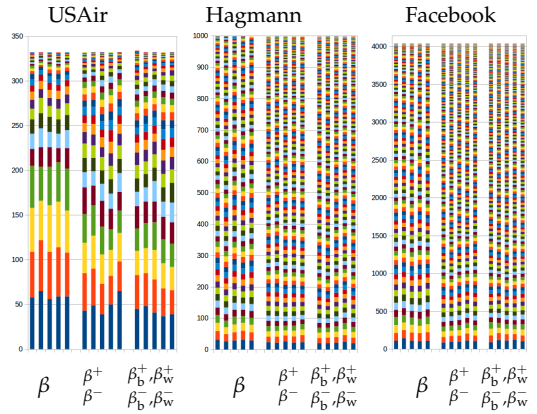


Fig. 2: Accumulative cluster sizes for five restarts for the three sampling procedures on the same training data. The results are based on the clustering inferred after the last sweep of the sampling procedure.

structural information about the network. This further strengthens the indication that IRM describes the structural properties of the network on a more detailed block-level without introducing additional overfitting to the training data.

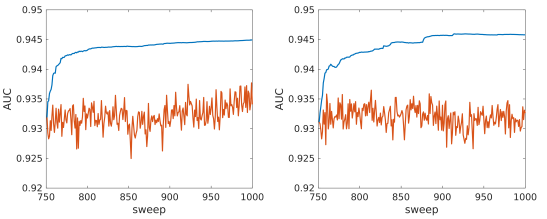


Fig. 3: The progress of AUC for the USAir network. For sweep i, the blue line shows AUC based on the average clustering for sweep 750 to i (Bayesian average). The red line shows AUC for the particular clustering at sweep i. Results are averaged for five restarts on the same hold out data.

Table 4 shows the average AUC computed when basing the link probabilities on the final clustering versus on the average of the last 250 sweeps of the sampling procedure. Using the Bayesian average performs significantly better. The effect is further illustrated in figure 3 for the USAir network.

Table 3 compares the inferred hyperparameters with the network density and the link density of the inferred

	Sampled parameters	AUC Sweep 1000	AUC Averaged
USAir	β	0.9341 ± 0.005	0.9449 ± 0.002
	β^+, β^-	0.9286 ± 0.007	0.9467 ± 0.003
	$\beta_b^+, \beta_b^-, \beta_w^+, \beta_w^-$	0.9327 ± 0.004	0.9458 ± 0.004
Hagmann	β	0.9112 ± 0.001	0.9127 ± 0.001
	β^+, β^-	0.9162 ± 0.001	0.9184 ± 0.001
	$\beta_b^+, \beta_b^-, \beta_w^+, \beta_w^-$	0.9164 ± 0.001	0.9186 ± 0.001
Facebook	β	0.9876 ± 0.000	0.9885 ± 0.000
	β^+, β^-	0.9878 ± 0.000	0.9890 ± 0.000
	$\beta_b^+, \beta_b^-, \beta_w^+, \beta_w^-$	0.9876 ± 0.000	0.9891 ± 0.000

Table 4: Comparing AUC, computed for the averaged clustering of the last 250 sweeps and computed for the last sweep. The results are the average for five different runs using a single hold out data set.

clusterings. This clearly indicates that sampling the hyperparameters reflects learning block level cluster densities, rather than simply reflecting the overall network link density.

4. CONCLUSION

We have investigated the influence of various hyperparameter constructions in the infinite relational model for clustering complex real world networks. We find that the hyper-parameter construction significantly influences the number of inferred components as well as the predictive performance of the model. We have demonstrated that using informed asymmetric priors can improve predictive performance compared to uninformed symmetric priors, and that the approach proposed in [3] assuming a symmetric prior $\beta = \beta^+ = \beta^-$ that is inferred was outperformed in link prediction by inferred asymmetric priors, providing a more refined block-structure. Separately sampling parameters for within and between components allowed the model to account for even more components without indications of overfitting to the training data. For the examined networks, sampling asymmetric hyper-parameters in IRM performs on par with using joint assymetric priors fixed to reflect the network density for an adequately chosen scale c. However, we find that inferring the hyper-parameters does not simply reflect the density of the network, but reflects the average link densities at the levels of the identified blocks which cannot be estimated in advance from the network.

5. REFERENCES

[1] Harrison C White, Scott A Boorman, and Ronald L Breiger, "Social structure from multiple networks. i. blockmodels of roles and positions," *American journal of sociology*, pp. 730–780, 1976.

[2] Paul W Holland, Kathryn Blackmond Laskey, and Samuel Leinhardt, "Stochastic blockmodels: First steps," *Social networks*, vol. 5, no. 2, pp. 109–137, 1983.

[3] Charles Kemp, Joshua B Tenenbaum, Thomas L Griffiths, Takeshi Yamada, and Naonori Ueda, "Learning systems of concepts with an infinite relational model," in *Proceedings of the national conference on artificial intelligence*, 2006, vol. 21, p. 381.

[4] Zhao Xu, Volker Tresp, Kai Yu, and Hans-Peter Kriegel, "Learning infinite hidden relational models," *Uncertainity in Artificial Intelligence (UAI2006)*, 2006.

[5] Tue Herlau, Mikkel N Schmidt, and Morten Mørup, "Infinite-degree-corrected stochastic block model," *Physical Review E*, vol. 90, no. 3, pp. 032819, 2014.

[6] Sonia Jain and Radford M Neal, "A split-merge markov chain sampling algorithm for bayesian mixture models," *Journal of Computational and Graphical Statistics*, vol. 13, no. 1, pp. 158–182, 2004.

[7] Vladimir Batagelj and Andrej Mrvar, "Pajek datasets," *http://vlado.fmf.uni-lj.si/pub/networks/data/*, 2006.

[8] Linyuan Lü, Liming Pan, Tao Zhou, Yi-Cheng Zhang, and H Eugene Stanley, "Toward link predictability of complex networks," *Proceedings of the National Academy of Sciences*, vol. 112, no. 8, pp. 2325–2330, 2015.

[9] Patric Hagmann, Leila Cammoun, Xavier Gigandet, Reto Meuli, Christopher J Honey, Van J Wedeen, and Olaf Sporns, "Mapping the structural core of human cerebral cortex," *PLoS Biology*, vol. 6, no. 7, pp. 1479–1493, 2008.

[10] Julian J McAuley and Jure Leskovec, "Learning to discover social circles in ego networks.," in *NIPS*, 2012, vol. 2012, pp. 548–56.

[11] Jure Leskovec and Andrej Krevl, "Snap datasets: Stanford large network dataset collection," *https://snap.stanford.edu/*.

[12] Jin Huang and Charles X Ling, "Using auc and accuracy in evaluating learning algorithms," *Knowledge and Data Engineering, IEEE Transactions on*, vol. 17, no. 3, pp. 299–310, 2005.

[13] Kurt Miller, Michael I Jordan, and Thomas L Griffiths, "Nonparametric latent feature models for link prediction," in *Advances in neural information processing systems*, 2009, pp. 1276–1284.

[14] David Liben-Nowell and Jon Kleinberg, "The link-prediction problem for social networks," *Journal of the American society for information science and technology*, vol. 58, no. 7, pp. 1019–1031, 2007.

[15] Aaron Clauset, Cristopher Moore, and Mark EJ Newman, "Hierarchical structure and the prediction of missing links in networks," *Nature*, vol. 453, no. 7191, pp. 98–101, 2008.

Appendix B

Numerical Approximations for Speeding Up MCMC Inference in the Infinite Relational Model

Numerical Approximations for Speeding Up MCMC Inference in the Infinite Relational Model. Mikkel N. Schmidt and Kristoffer Jon Albers. In European Signal Processing Conference (EUSIPCO), 2015.

NUMERICAL APPROXIMATIONS FOR SPEEDING UP MCMC INFERENCE IN THE INFINITE RELATIONAL MODEL

Mikkel N. Schmidt and Kristoffer Jon Albers

Cognitive Systems, DTU Compute
Technical University of Denmark
Richard Petersens Plads, DTU Bldg. 321.
2800 Lyngby, Denmark.

ABSTRACT

The infinite relational model (IRM) is a powerful model for discovering clusters in complex networks; however, the computational speed of Markov chain Monte Carlo inference in the model can be a limiting factor when analyzing large networks. We investigate how using numerical approximations of the log-Gamma function in evaluating the likelihood of the IRM can improve the computational speed of MCMC inference, and how it affects the performance of the model. Using an ensemble of networks generated from the IRM, we compare three approximations in terms of their generalization performance measured on test data. We demonstrate that the computational time for MCMC inference can be reduced by a factor of two without affecting the performance, making it worthwhile in practical situations when on a computational budget.

***Index Terms*—** Nonparametric Bayesian modeling, Infinite Relational Model, Numerical approximation.

1. INTRODUCTION

A common approach to modeling the structure in complex network data is to cluster the nodes of the network into groups which have similar structural properties. Discovering groups of nodes which connect to other nodes in a similar fashion is useful for unsupervised, explorative analysis of complex networks. Using non-parametric Bayesian models, one can find cluster structure which is statistically salient and learn the apropriate number of clusters at the same time.

Many different non-parametric Bayesian models of complex networks exist in the literature. The arguably simplest model is the socalled infinite relational model (IRM) [1–3], which is a non-parametric Bayesian extension of the stochastic blockmodel [4, 5]. Since exact inference in the IRM is intractable for networks with more than a few nodes, the clustering is typically learned using approximate inference techniques such as Markov chain Monte Carlo (MCMC) [6], or variational Bayes [7].

When analyzing very large complex networks, the computational speed of the inference procedure can become an issue [8]. There are, at least, four different ways in which one might consider speeding up the inference procedure when performing cluster analysis of complex networks. i) The model can be simplified in a way to permit analytic solutions to part of the problem or in other ways achieve faster inference. ii) Approximate inference algorithms with a better speed-accuracy trade-off can be employed. iii) Computational optimizations such as parallelization, vectorization, and lookup tables can be implemented. iv) Numerical approximations can be computed at a lower precision.

In this paper we investigate how low-precision numerical approximations can be used to speed up MCMC inference in the infinite relational model. We examine how changes in the numerical precision affects the inferred clustering structure. In particular we test different numeric approximations to the evaluation of the log-gamma function, which consitutes the majority of the work in a Gibbs sampler for the IRM.

2. THE INFINITE RELATIONAL MODEL

The IRM extends the stochastic blockmodel, by relying on a nonparametric prior over partitions to flexibly allow the number of clusters in the model to be learned from the data. This allows the model to adapt to the size and complexity of the data.

Restrict the discussion to the modeling of simple unipartite networks, the IRM can be formulated as the following generative process,

$$z|\alpha \sim \mathrm{CRP}(\alpha), \tag{1}$$

$$\theta_{k,\ell}|a,b \sim \mathrm{Beta}(a,b), \tag{2}$$

$$A_{i,j}|\theta,z \sim \mathrm{Bernoulli}(\theta_{z_i,z_j}), \tag{3}$$

where $A_{i,j}$ is a binary variable indicating whether or not there exists a link between node i and j. The prior for the cluster assignment, z is a Chinese restaurant process (CRP) governed

by the concentration parameter α,

$$p(z|\alpha) = \frac{\Gamma(\alpha)\alpha^K}{\Gamma(\alpha+N)} \prod_{k=1}^{K} \Gamma(m_k), \tag{4}$$

where N is the number of nodes, K is the number of clusters, and m_k is the number of nodes in cluster k. The probability of observing a link between two nodes i and j, follows a Bernoulli distribution, depending only on z and the parameters $\theta_{k,\ell}$ which specifies the link probability between nodes in the two clusters k and ℓ, that i and j are assigned to. A Beta distribution with parameters a and b is used as a prior for these link probabilities,

$$p(\theta_{k,\ell}|a,b) = \frac{\theta_{k,\ell}^{a-1}(1-\theta_{k,\ell})^{b-1}}{B(a,b)}. \tag{5}$$

Due to the conjugacy of the Bernoulli likelihood and Beta prior, the parameters θ can be analytically marginalized yielding the following likelihood,

$$p(A|z,a,b) = \prod_{k=1}^{K} \prod_{\ell=k+1}^{K} \frac{\mathrm{B}(m_{k,\ell}+a, \bar{m}_{k,\ell}+b)}{\mathrm{B}(a,b)}, \tag{6}$$

where $m_{k,\ell}$ and $\bar{m}_{k,\ell}$ denote the number of links and non-links between nodes in cluster k and ℓ, and B denotes the Beta function,

$$\mathrm{B}(x,y) = \frac{\Gamma(x)\Gamma(y)}{\Gamma(x+y)}. \tag{7}$$

In the following we will keep α constant and assume improper flat priors over a and b.

2.1. Computations for a Gibbs sampler

The computations required to implement a Gibbs sampler for the cluster assignments z can be found by considering the change in the likelihood (and prior) when moving a node to each of the existing clusters or to a new cluster. Reassigning node n to the cluster k will change the likelihood by the factor

$$\prod_{\ell} \frac{\mathrm{B}(m_{k,\ell}^{\setminus n} + r_{n,\ell} + a,\ \bar{m}_{k,\ell}^{\setminus n} + n_\ell - r_{n,\ell} + b)}{\mathrm{B}(m_{k,\ell}^{\setminus n} + a,\ \bar{m}_{k,\ell}^{\setminus n} + b)}, \tag{8}$$

where the count statistics $m_{k,\ell}^{\setminus n}$ and $\bar{m}_{k,\ell}^{\setminus n}$ are the number of links and nonlinks between cluster k and ℓ, ignoring node n, and $r_{n,\ell}$ is the number of links between n and all nodes in cluster ℓ (see [3] for details.) The count statistics for links and nonlinks between clusters can be kept and updated, instead of recomputed for each Gibbs iteration, and thus the required computations for implementing the Gibbs sampler is dominated by evaluating the Beta function.

In a practical implementation, computations will be performed in the log domain in order to ensure numeric stability. Calculating the logarithm of the Beta function is therefore the central operation for evaluating the likelihood within the Gibbs sampler. Thus, it is important to have efficient means for computing the logarithm of the Gamma function, in order to efficiently compute the logarithm of the Beta function,

$$\log \mathrm{B}(x,y) = \log\Gamma(x) + \log\Gamma(y) - \log\Gamma(x+y). \tag{9}$$

2.2. Computational optimization

In some situations, computing the log-Gamma function can be completely avoided by computational optimization. From Eq. (8) it is evident, that if a and b are constant, the log-Gamma function need only be evaluated at integer steps ($I + a$, $I + b$, and $I + a + b$, for integer I). Thus, it might be practical to simply precompute a large lookup table of log-Gamma values. However, depending on the data, the required lookup table might be impractically large, and if a and b are allowed to vary, computing a large lookup table before each Gibbs sweep is not practical.

2.3. Approximation by maximization

The reason that the Beta function arises is the analytical marginalization of θ. The joint distribution of θ and the data for a single block (all links and non-links between nodes in cluster k and ℓ) is given by

$$p(A_{k,\ell}, \theta|z,a,b) = \frac{\theta^{m+a-1}(1-\theta)^{\bar{m}+b-1}}{\mathrm{B}(a,b)}, \tag{10}$$

where, to simplify the exposition, we have omitted the k,ℓ subscripts in the parameter and count statistics. Marginalizing θ yields the term $\mathrm{B}(m+a, \bar{m}+b)$. A crude approximation is to replace the marginalization by plugging in the maximum a-posteriori (MAP) estimate of θ, which yields

$$\frac{(m+a-1)^{m+a-1}(\bar{m}+b-1)^{\bar{m}+b-1}}{(n+a+b-2)^{n+a+b-2}}. \tag{11}$$

Taking the logarithm and comparing with Eq. (9) we see that the MAP-plugin estimate corresponds exactly to approximating the log-Gamma function using Stirling's approximation (the variant for the factorial function), $\log n! = \log\Gamma(n) \approx n\log(n) - n$. This gives some understanding as to what happens algorithmically when approximating the log-Gamma function in this manner.

2.4. Directly approximating the log-Gamma function

Many well-known approximations to the log-Gamma function exist, having different trade-off between computational complexity and precision (see Fig. 1). One of the simplest is Stirling's approximation, given by

$$\log\Gamma(x) \approx \tfrac{1}{2}\log(2\pi) + (x-\tfrac{1}{2})\log(x) - x. \tag{12}$$

Stirling's approximation is relatively fast to compute, as it only involves a single logarithm and a multiplication (disregarding additions and computing the constant in advance.) Stirling's approximation yields an asymptotically accurate approximation, but is not very precise for small values of its argument.

We have discovered a very similar approximation with the same computational complexity,

$$\log\Gamma(x) \approx 1 + (x - 2 + \tfrac{1}{\log(2)})\log(x) - x. \quad (13)$$

This approximation is not an asymptotic formula, but it has better precision for small values, $x < 4$, and thus a better worst case error. As we have not been able to find this approximation in the literature, we refer to it in the following as Schmidt's approximation.

Gosper's approximation,

$$\log\Gamma(x) \approx 1 + \tfrac{1}{2}\log([2x - \tfrac{5}{3}]\pi) + (x-1)\log(x-1) - x, \quad (14)$$

involving two logarithms and two multiplications, yields better asymptotic behavior but is still very imprecise for low values.

Lanczos' family of approximations [9] include the particularly interesting $\gamma = 1.5$ approximation with only two terms, which we refer to as Lanczos 1.5,

$$\log\Gamma(x) \approx \tfrac{1}{2}\log(2\pi) + (x + \tfrac{1}{2})\log(x+1) - (x+1) + \log(c_0 + \tfrac{c_1}{x}), \quad (15)$$

where c_0 and c_1 are constants. It involves two logarithms, one multiplication, and one division but yields a very good precision also for small values, having a relative error of no more than $2.4 \cdot 10^{-4}$ [9]. By increasing the number of terms in the Lanczos' approximation to N (requiring two logarithms, one multiplication and N divisions) the log-Gamma function can be computed to arbitrary precision.

2.5. Approximating the logarithm

Since the discussed approximations of the log-Gamma functions still require the computation of one or more logarithms, which itself requires evaluating some series expansion, nothing much appears to be gained. However, the logarithm can be approximated very cheaply. In most computer systems, floating point numbers are represented by a sign (S), a mantissa (M), and an exponent (E),

$$x = (-1)^S + M \cdot 2^E, \quad (16)$$

where $0.5 \leq M \leq 1$. For positive x, the logarithm is given as

$$\log(x) = \log(M) + \log_2(e)^{-1} E. \quad (17)$$

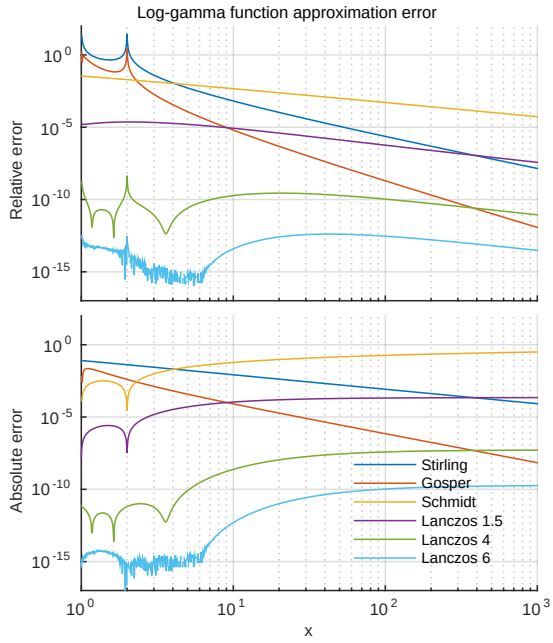


Fig. 1. Relative and absolute error in approximating $\log\Gamma(x)$ using various methods.

Thus, we need only one multiplication and the computation of the logarithm of the mantissa. Since the mantissa by definition is between one half and one, its logarithm can be approximated very efficiently.

Classical expansions, such as the rational expansion

$$\log(x) \approx 2\left(r + \frac{r^3}{3}\right), \quad r = \frac{x-1}{x+1}, \quad (18)$$

and the Taylor series (at a)

$$\log(x) \approx \log(a) + \frac{x-a}{a} - \frac{(x-a)^2}{2a^2} + -\frac{(x-a)^3}{3a^3} - \cdots, \quad (19)$$

are not particularly precise when using a low order (see Fig 2.) Approximating the logarithm direcly using lookup table is also not very precise, even for quite large tables. However, creating a lookup table of Taylor expansions yields very fast and precise approximations. For example, using a table with 1024 first order Taylor approximations yield a relative error below 10^{-5} and requires only one lookup and one division. Preferably, the division can be changed to a lookup and a multiplication by storing precomputed values of $\frac{1}{a}$.

The number of elements in the lookup table can be decided in advance, and by keeping the lookup table small enough, the entire is likely to fit the CPU cache at runtime, avoiding expensive access to the main memory.

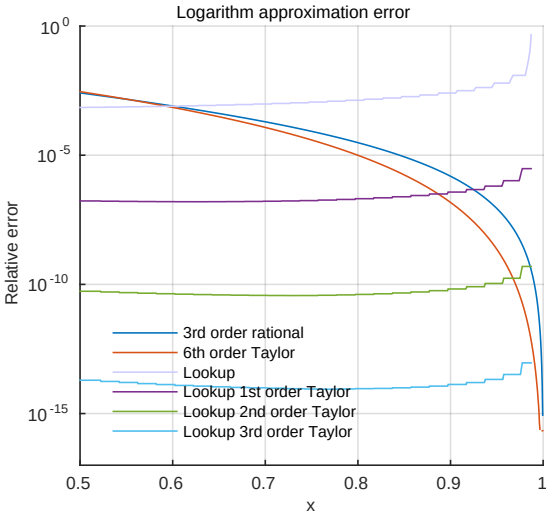


Fig. 2. Relative error in approximating $\log(x)$. The Taylor series shown is expanded around $a = 1$, and all lookup tables are of size 1024.

3. EXPERIMENTAL EVALUATION

To examine the influence of different approximation strategies in the IRM, we conducted an empirical evaluation of the performance of the model on a set of generated networks. We created 1000 random networks sampled from the IRM with 50 nodes, and parameters $a = b = \alpha = 1$. For each of the 1000 networks we generated z and θ from the prior and then sampled two network realizations, using one set for training and one set for testing.

To evaluate and compare the influence of the numeric approximations on the performance of IRM, the model was run on the same data with three different ways of approximating the log-Gamma function as well as with the log-Gamma computed to machine precision. In all approximations we computed logarithms using a length 1024 lookup table of first order Taylor expansions.

For each network, the model was fitted using 10,000 Gibbs sweeps over the clustering z interleaved with 100,000 Metropolis-Hastings (MH) updates of a and b. We repeatedly conducted 10 Gibbs sweeps followed by 100 MH updates, thinning the MCMC sample by a factor of 10. The proposal distribution for the MH update was a Normal distribution with standard deviation 0.1.

To compare the generalization performance, we computed the posterior mean log-likelihood averaged over the 1000 test networks. For each iteration number, we averaged over all previous MCMC samples in order to evaluate the test log-likelihood as a function of the number of iterations (see Figure 3.)

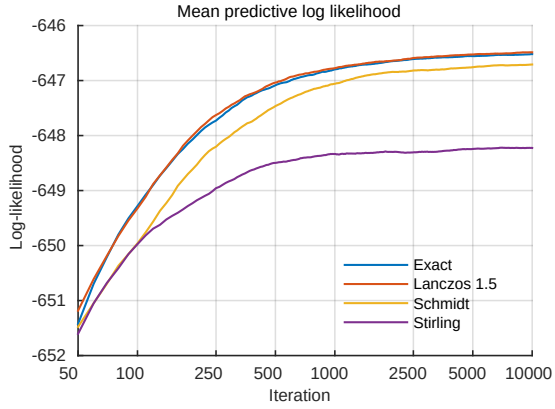


Fig. 3. Average test log-likelihood of the inferred model as a function of the number of iterations (Gibbs sweeps) of the MCMC sampler.

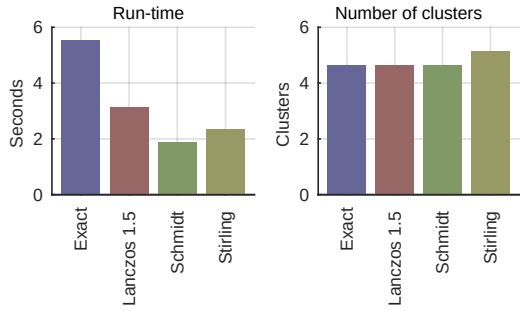


Fig. 4. Performance of the model using the different approximations, in terms of the average computation time and estimated number of clusters.

We recorded only the time it took to conduct the Gibbs sweeps, disregarding time for loading data, conducting MH updates, and storing intermediate results etc.

3.1. Results

Figure 3 shows that the performance of the Lanczos 1.5 approximation is indistinguishable from the exact computation. The performance of Schmidt's approximation is close but significantly worse, and Stirling's approximation performs much worse.

Figure 4 shows the run-time as well as the number of clusters discovered. For comparison, we note that the average number of clusters in the ensemble of networks used for training and test is $\alpha\psi(\alpha + n) - \psi(\alpha) \approx 4.49$. As expected, the three approximations are significantly faster than the machine precision computations: Lanczos 1.5 is around 2 times faster, and Schmidt is around 3 times faster. Although Stirling's ap-

proximation has the same complexity as Schmidt's, it runs slower: This can be explained by examining the number of clusters discovered by the different algorithms. While Lanczos 1.5 and Schmidt find almost exactly the same number of clusters as the exact computations, Stirlings approximation leads the inference procedure to erroneously discover more clusters which in turn impacts the computation time, making Stirlings approximation both less accurate and slower.

4. CONCLUSIONS

Introducing numerical approximation in evaluating the likelihood of a Bayesian model will influence the inference. In non parametric models, the inferred model complexity depends on the complexity of the data. Hence, introducing numerical approximations will likely affect the inferred complexity of the model making it difficult to assess the repercussions of the approximation compared to parametric models.

In our experiments we observed that the inferred number of components in the IRM depends on the chosen approximation of the log-Gamma function. Stirling's approximation is very imprecise for small values of its arguments, and using this approximation introduces bias in the model, which turns out to have a significant influence on the performance of the model: The MCMC procedure converges, but to solutions with a lower test likelihood and with notably more clusters. The proposed Schmidt's approximation is more precise in the low range, and though it does not appear to overestimate the number of components it converges to a lower test likelihood than the exact computation. The Lanczos 1.5 approximation on the other hand appears to yield results indistinguishable from the exact computation at around half the computational cost.

There appear to be no reason not to use the Lanczos 1.5 approximation in practical applications of IRM analyses, when one is on a tight computational budget. If one is willing to accept some error in approximation, Schmidt's approximation is to be preferred over Stirling's approximation.

In a practical implementation it is likely to be beneficial to use a hybrid approach, e.g. using a Lanczos approximation for small values, while relying on Gosper's approximation (for better precision) and/or Stirling's approximation (for lower computation complexity) for larger values. As mentioned, alternative to the approximations discussed here, one should also consider if it is more efficient to simply precompute a lookup table of the needed log-Gamma value: This, however, also depends on the data—both on the size of the network, the number of links, and the number of inferred clusters.

Acknowledgement: This project was supported by the Lundbeck Foundation, grant nr. R105-9813.

REFERENCES

[1] Charles Kemp, Joshua B Tenenbaum, Thomas L Griffiths, Takeshi Yamada, and Naonori Ueda, "Learning systems of concepts with an infinite relational model," *Cognitive Science*, vol. 21, no. 1, pp. 381, 2006.

[2] Zhao Xu, Volker Tresp, Kai Yu, and Hans-Peter Kriegel, "Infinite Hidden Relational Models," in *Proceedings of the 22nd International Conference on Uncertainity in Artificial Intelligence*, 2006.

[3] Mikkel N. Schmidt and Morten Mørup, "Nonparametric Bayesian modeling of complex networks: An introduction," 2013.

[4] Tom A B Snijders and Krzysztof Nowicki, "Estimation and prediction for stochastic blockmodels for graphs with latent block structure," *Journal of Classification*, vol. 14, no. 1, pp. 75–100, 1997.

[5] P W Holland, K B Laskey, and S Leinhardt, "Stochastic blockmodels: First steps," *Social Networks*, vol. 5, no. 2, pp. 109–137, 1983.

[6] Kristoffer Jon Albers, Andreas Leon Aagard Moth, Morten Mørup, and Mikkel N. Schmidt, "Large scale inference in the Infinite Relational Model: Gibbs sampling is not enough," in *IEEE International Workshop on Machine Learning for Signal Processing, MLSP*, 2013.

[7] Katsuhiko Ishiguro, Issei Sato, and Naonori Ueda, "Collapsed Variational Bayes Inference of Infinite Relational Model," 2014.

[8] Karen S Ambrosen, Tue Herlau, Tim Dyrby, Mikkel N Schmidt, and Morten Mørup, "Comparing Structural Brain Connectivity by the Infinite Relational Model," in *Pattern Recognition in NeuroImaging (PRNI)*, 2014.

[9] C. Lanczos, "A Precision Approximation of the Gamma Function," *Journal of the Society for Industrial and Applied Mathematics*, vol. 1, pp. 86–96, 1964.

Appendix C

Local Modes in the Posterior Distribution of Dirichlet Process Mixture Models

Local Modes in the Posterior Distribution of Dirichlet Process Mixture Models. Kristoffer Jon Albers, Morten Mørup, and Mikkel N. Schmidt. (preliminary work).

Local modes in the posterior distribution of Dirichlet process mixture models

Kristoffer Jon Albers **Morten Mørup** **Mikkel N. Schmidt**
Section for Cognitive Systems, DTU Compute, Technical University of Denmark

Abstract

Learning a clustering in a mixture model can be difficult because of local modes which may trap the learning algorithm. In particular, Dirichlet process mixture models, in which the number of mixture components is learned from data can be challenging in this respect. In this paper we investigate and characterize the local modes of the posterior distribution for a particular relational Dirichlet process mixture model for network data. We count, visualize, and characterize the local modes in relation to the data. Our result show that even in small datasets the number of local modes is staggering. The local modes that are easy to find are often good but not the best and often substantially different from the global optimum. And although there are very many local modes which yield good predictive performance, there are only a few that are excellent approaching the performance of the Bayes average, underlining that local modes are an important issue that should be understood and handled appropriately.

1 Introduction

Dirichlet process (DP) mixture models are a popular Bayesian nonparametric generalization of finite mixture models, useful for finding latent cluster structure while adapting the number of clusters to the complexity of the data. Mixture models are known to suffer from problems with local optima, which can be problematic for any learning algorithm which aim to learn a clustering, whether the algorithm seeks to optimize or average over the space of partitions. When the learning algorithm is caught in a local solution, it can lead to sub-par performance, and the lack of global convergence can be very difficult to assess. In practice, techniques such as annealing and multiple random restarts are often used to protect against getting caught in local optima.

In this paper we investigate and characterize the posterior landscape of a DP mixture model in terms of its local optima, and examine its characteristics for varying data size. We define a local optimum as a clustering from which no single node can be reassigned to an existing or a new empty cluster, without a decrease in the posterior probability of the resulting clustering.

We focus on a particular relational Dirichlet process mixture model for network data called the infinite relational model (IRM) [14]. In its most simple form, the IRM models a network (a simple graph) using a Bernoulli likelihood and conjugate Beta priors, which allows the analytic marginalization of all parameters except the clustering of the network nodes. (Many extensions of the IRM have been proposed in the literature, see e.g. [13] for a review.) We believe this simple model, parameterized only by a set partition, is ideal to illustrate the local optima of the posterior distribution, but note that other conjugate models, such as a Gaussian DP mixture model, could equally well have served to illustrate our points.

We base our discussion on inference using Markov Chain Monte Carlo (MCMC) methods [18]. Monte Carlo simulations can be used to estimate the posterior, if samples are drawn from the correct distribution, which can be achieved by basing the sampling on a Markov Chain with the posterior as its stationary distribution. In practical applications, it might be difficult to reach the stationary distribution, for various reasons:

- Though methods for convergence assessment do exist, it can be difficult to determine whether a MCMC algorithm has in fact explored the majority of the state space [8, 7].
- MCMC-sampling is prone to get stuck in local suboptimal regions [19, 6], from where it is unlikely to reach the global region with the highest probability mass.
- For large, complex problems, computational limitations can make it unrealistic to expect the sampler to explore most regions of the state space [20].

PRELIMINARY WORK

Our practical experience with using MCMC to fit mixture models to large real world datasets suggest that often, getting caught in a local optimum is not disastrous: A local optimum can give a sufficiently good predictive performance in a practical setting. This motivates us to study how getting stuck in local optima influences the performance of the model: Are we looking for a needle in the haystack? Is there one or perhaps a few local optima that are superior, or is there a large number of local optima with almost equal performance?

In this paper we will investigate the practical implications of local optima by addressing the following questions:

- How many local optima exist, and how does the number vary with the size and complexity of the network?
- How different are the local solutions that can be found in real world networks?
- The "basins of attraction" of local optima might be substantially different, which means that some are easier to find than others. How does this co-vary with the quality of the local solution?
- How do local solutions compare to the global solution and the Bayes optimal average in terms of structure and predictive performance?

To answer these questions, we first examine the small network setting, where all local optima can be found by brute force computation. Next, we analyze 12 real networks of varying size and compare with an ensemble of synthetic random networks. To assess the influence of local optima on predictive performance we examine a large number of synthetic networks generated from the IRM model, for which independent hold-out test data is available. Finally, we graphically illustrate the posterior landscape of a real network using a latent embedding method. But first, we set the stage by reviewing the IRM.

2 The infinite relational model

A mixture model is a probabilistic model often used for clustering. It models a probability distribution as a mixture of simpler distributions, distributing data into a number of components, by which the latent classes arising from data dependencies are identified [9]. While the number of components that best captures the latent classes depends on the data complexity, the simple finite mixture model assumes that data is separable into a known number of clusters. Hence it often becomes a challenging task to select a model with appropriate complexity to fit the observed data. A viable alternative is to use a Bayesian nonparametric generalisation of the finite mixture model. The Bayesian approach allows for combining prior information with the observed data while the nonparametric approach allows for estimating both individual component parameters as well as an appropriate number of components. Using the Dirichlet process as prior over the mixture distribution we obtain a Dirichlet process mixture model [10, 11], one of the most popular Bayesian nonparametric models [12].

The stochastic block model (SBM) [16, 3] forms a simple mixture model for data in the form of a graph. We consider only simple graphs with N nodes and L undirected, unweighted links. The SBM assumes that each node can be assigned to some unknown class, and that all links in the network are independent given the class labels. The infinite relational model (IRM) [14, 15] extends the SBM from a finite to a DP mixture by introducing a Chinese restaurant process (CRP) [17] prior for the latent partitioning z of the nodes.

$$z \sim \mathrm{CRP}(\alpha), \qquad p(z|\alpha) = \frac{\Gamma(\alpha)\alpha^K}{\Gamma(\alpha+N)} \prod_{k=1}^{K} \Gamma(m_k), \tag{1}$$

where m_k is the number of nodes in cluster k. Each link $A_{i,j}$ in the graph is generated independently from a Bernoulli distribution where the parameter depends only on the class labels of the links.

$$A_{i,j} \sim \mathrm{Bern}(\theta_{z_i,z_j}), \quad p(A_{i,j}|\theta,z) = \theta_{z_i,z_j}^{A_{i,j}}(1-\theta_{z_i,z_j})^{1-A_{i,j}} \tag{2}$$

The probability of observing a link between nodes in cluster k and ℓ is thus given by $\theta_{k,\ell}$ which is assigned a Beta prior distribution.

$$\theta_{k,\ell} \sim \mathrm{Beta}(a,b), \quad p(\theta_{k,\ell}|a,b) = \frac{\theta_{k,\ell}^{a-1}(1-\theta_{k,\ell})^{b-1}}{\mathrm{B}(a,b)} \tag{3}$$

Since the Beta prior is conjugate to the Bernoulli likelihood, the probabilities $\theta_{k,\ell}$ of links between each pair of clusters can be analytically marginalized, yielding the following effective likelihood

$$p(A|z,a,b,\alpha) = \prod_{k=1}^{K} \prod_{\ell=k+1}^{K} \frac{\mathrm{B}(m_{k,\ell}+a, \bar{m}_{k,\ell}+b)}{\mathrm{B}(a,b)}, \tag{4}$$

where K is the number of clusters, and $m_{k,\ell}$ and $\bar{m}_{k,\ell}$ are the number of links and non-links between clusters k and ℓ.

3 Exact analysis of local optima

For networks with few enough nodes, it is computationally feasible to find all local optima for all possible networks. This can aid in revealing the relation between the topology of the network and number of local optima, although this might not generalize to larger networks.

The total number of possible partitions of a set of size N is given by the Bell number B_N [5], the first ten terms of which is 1, 2, 5, 15, 52, 203, 877, 4140, 21147, 115975 (starting with B_1). All set partitions can be enumerated using a simple algorithm [23], thus for a small network it is

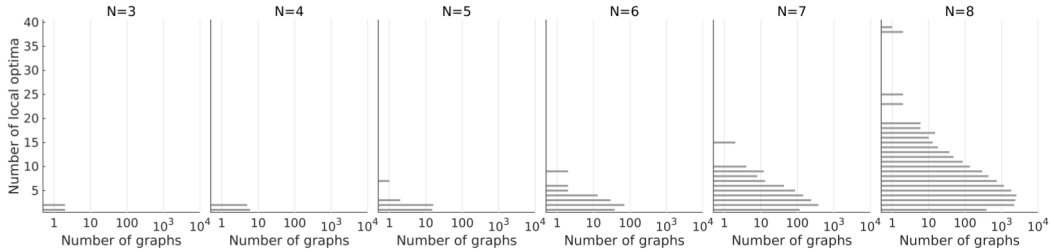


Figure 1: Distribution of the number of local optima for all for all non-isomorphic graphs with 3–8 nodes.

feasible to examine each clustering to check if it is a local optimum. We need only consider the set of non-isomorphic networks, the number of which is given by I_N [4] where the first ten terms are 1, 2, 4, 11, 34, 156, 1044, 12346, 274668, 12005168. Algorithms and software for enumerating all non-isomorphic networks is available [2].

Using this approach we counted the number of local optima in all networks up to $N = 8$. (A better optimized implementation might enable us to go up to 9 or 10, but probably not much higher than that: For $N = 11$ the number of combinations is in the trillions). The number of local optima is shown in Fig. 1. As might be expected, the average number of local optima increases with the network size, but we note also that for $N > 5$ a few networks have particularly many local optima. Examining these outliers, we see that they are networks with particularly strong symmetries. For example, the 5-node network with most local optima is a ring, ⋱. The nodes can be clustered to form the following seven local optima:

Just as the 5-node network ⋱ is topologically isomorphic to the above ring, it is clear that clustering 3 - 7 are identical if the nodes are relabeled. Similarly, the three 8-node networks with most local optima (38–39) are highly symmetric,

and the high number of local optima is again due to a large number of identical clusterings under a relabeling of the nodes.

4 Local optima for large networks

For larger networks it is not possible to investigate all possible cluster configurations, to find the ensemble of local optima. Instead we will estimate local optima from a representative set of cluster configurations. This set is either obtained by a set of uniformly random [22] cluster configurations or by using a MCMC procedure to explore the parameter space, and find the local optimum for each visited cluster configuration.

4.1 MCMC inference

Inferring the posterior distribution $p(z|A, a, b, \alpha)$, is not possible in an exact analytical form. Instead, Gibbs sampling can be used [18], by evaluating the posterior conditional distribution of reassigning the nodes one at a time. A node n is reassigned to some cluster ℓ according to the following probability distribution:

$$p(z_n = \ell|A, z_{\setminus n}, a, b, \alpha) = \frac{p(A, z_{\setminus n}, z_n = \ell|a, b, \alpha)}{\sum_{k=1}^{K+1} p(A, z_{\setminus n}, z_n = k|a, b, \alpha)} \quad (5)$$

where $z_{\setminus n}$ indicates that node i is ignored, and the summation is over assigning n to each of the existing clusters as well as a new, empty cluster.

As the Gibbs sampler only evaluates one node at a time, it is prone to get stuck in local optima and strugles to explore the space for more complex problems [19, 20]. More sophisticated sampling strategies are therefore essential. One is the split merge procedure, as proposed in [21], which evaluates more nodes at a time by proposes of splitting a single cluster in two, or merging two clusters to one.

4.2 Local posterior optimizer

To find local optima, we can use a similar procedure as the Gibbs sampler, by iteratively evaluating reassignments of one node at a time. But instead of randomly reassign the node according to the probabilities in 5, the node will deterministically be assigned to the cluster with the highest probability. This will transform the Gibbs sampling into a hill climbing optimization algorithm. While the Gibbs sampler will search the entire space, this algorithm has no mechanism for escaping local regions. It will stop when it reaches a local optimum, from where no single node can be reassigned to yield a better posterior.

Network	Nodes	Links	Description
Thurm	15	33	Social network formed from multiplex relations between employees in an office [26][28].
Sampson	25	74	Social network for novices in a monastery [25][29]. Binarized for mutual positive arcs.
Galesburg	31	39	Social network of friendship ties between physicians [25][30].
Karate	34	78	Social network between members in a karate club [24][32].
Dolphins	62	159	Social network of dolphin associations [24][31].
AdjNoun	112	425	Adjacency network of common nouns and adjective in "David Copperfield" [24][33].
PolBooks	105	441	Network of books about US politics sold through Amazon.com, compiled by Krebs [24].
FoodWeb	128	128	Food web from Florida Bay documented in the rainy season[36], as in [27].
Jazz	198	2742	Social network of collaboration between jazz musicians[37], as in [27].
Neural	297	2148	Neural network of the C.elegans roudworm[34]. Directions and weights ignored, as in [27].
USAir	332	2126	Traffic network of airlines in the US. Treshold to 2126 binary links, as in [27].
Metabolic	453	2025	Metabolic network of the C.elegans roundworm, as in [27].

Table 1: Dataset of 12 real world networks.

4.3 Data and experiments

We use both real and synthetic network, that are all undirected and unweighted. We use the 12 real world datasets presented in the Table 1. We use two datasets of synthetic networks. One consisting of 57 networks of sizes $N = 3, ..., 60$. These networks are randomly generated uniform, with a 50 percent chance of an edge between any pair of nodes. The other dataset consists of 500 random networks sampled from the IRM. Each network has with 50 nodes and hyperparameters $a = b = \alpha = 1$. For each network z and θ is generated from the prior and two network realizations are sampled, one used for training and one used for testing.

The following experiments were performed:

- To assess the number of local optima and the size of their basins of attraction, we computed the local optima found by using the hill climbing algorithm with 100,000 random initial cluster configurations.
- To approximate the posterior distribution, we did 100,000 iterations of MCMC sampling of, each consisting of performing one Gibbs sweep over all nodes followed by 100 split merge proposals. Five restarts of this strategy were performed for real world network, while a single restart was performed for each of the 500 synthetic networks.
- To find the best local optima (approximately find the global optimum), for each iteration of the above MCMC strategy, we found the nearest local optimum using the hill climbing algorithm.

In all experiments, the hyperparameters were kept constant, $\alpha = a = b = 1$. Table 2 shows the mean log posterior for the three experiments on the 12 real networks.

Network	Mean log posterior		
	Optimizer	*MCMC*	*MCMC + Optimizer*
Thurm	−62.4	−64.5	−60.9
Sampson	−124.6	−124.9	−120.8
Galesburg	−147.8	−143.6	−141.3
Karate	−201.6	−199.1	−196.2
Dolpins	−534.3	−528	−521.4
AdjNoun	−1429.8	−1427.3	−1411.8
Polbooks	−1281.6	−1253.5	−1249.2
FoodWeb	−3270.7	−3228	−3224.9
Jazz	−4785.6	−4684.2	−4680.9
Neural	−6986.6	−6839.3	−6824.4
USAir	−4988.8	−4883.3	−4863.9
Metabolic	−7478.8	−7030.5	−7011.5

Table 2: Mean log posterior for all clusterings encountered in the three experiments for the 12 real networks.

5 Results and discussion

Number of local optima Fig. 2 shows the found number of unique local optima as function of network size and number of edges. The number increases very quickly, and for networks with more than approximately 30 nodes / 100 edges starts to level off because we sample only 10^5 random initializations. Thus, we see that networks of this relatively small size have tens of thousands of local optima, and that the number increases super-exponentially with the data size.

The number of local optima of the real networks appears to follow the number of optima in the random graphs when considered as a function of the number of edges. When viewed by the number of nodes, the real networks have substantially fewer local optima than the random graphs, which is likely because the real networks are more sparse.

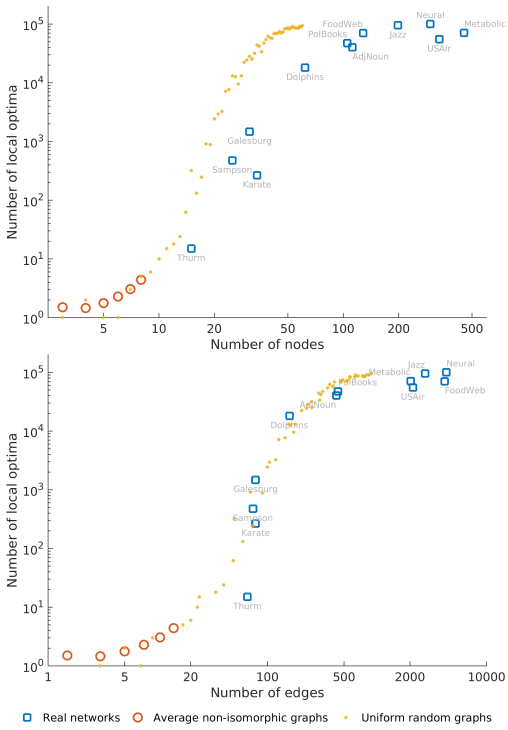


Figure 2: The number of distinct local optima found in 12 real networks by initializing in 10^5 uniformly random partitions and optimizing to the nearest local optimum. Results for 100 uniform random graphs and the average number of local optima in small nonisomorphic graphs are shown for comparison.

Although the real networks are assumed to have some inherent clustering structure, this does not seem to influence the number of local optima radically.

Basins of attraction Fig. 3 shows the posterior probability of each unique local optimum as a function of how often the optimum is found in the 10^5 random initializations. This illustrates the basins of attraction of the different local optima. For the five smallest networks there is a clear pattern: The most often found solution is among the those with the highest posterior probability, and the basin of attraction has clear positive correlation with the posterior probability. For the 7 largest networks there is a big variance in the posterior probability of the solutions that are not found often. In general (except in the Metabolic network) the solutions that are found most often tend to have high but not the best posterior probability. This fits with our practical experience that getting caught in a local optimum often is not disastrous, since the local optima with the largest basins of attraction in general are good (but clearly not the best) solutions.

Comparison of most often found and best optima Fig. 4 shows the best local optimum found and compares with the most often found local optimum, in order to assess the differences between the best and the most often found clusterings. For the smaller networks a few simple split/merge operations would be enough to move from the most often found to the best solution. For the larger networks, the necessary operation seems more complicated and affects up to 47% of the network nodes.

Predictive evaluation Fig. 6 shows the predictive performance measured on held-out data of the top 50 local optima found in the synthetic 50×50 networks, averaged over the 500 replications. The local optima are ranked both by their posterior probability and by their likelihood on the training data for comparison. Ranked by the posterior, the predictive performance of the best local optimum is much better than the second best, almost approaching the optimal performance of the Bayes average. Looking at the prior of the best clustering, it is clear that it has a strong influence, and that the best clustering has relatively fewer clusters than the subsequently ranked solutions. This makes intuitive sense: The more regularized solution leads to better predictive performance.

Ranked by the training likelihood, we see that there are a large number of solutions with almost equal posterior probability on the training set but which lead to sub-par predictive performance. This demonstrates how maximizing the likelihood is not a good strategy for this type of model.

Latent embedding visualization Fig. 5 illustrates the posterior landscape with its multitude of local optima for the Dolphins network.

Acknowledgements

This project was supported by the Lundbeck Foundation, grant nr. R105-9813.

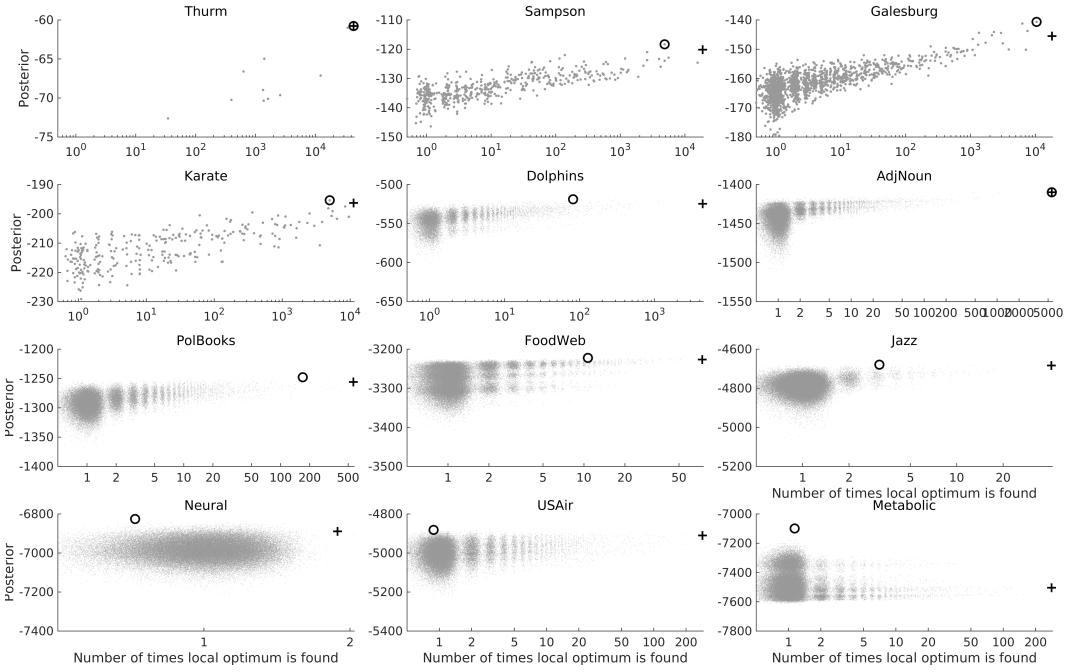


Figure 3: Posterior probability versus the probability of finding local optima. The plus indicates the local optima that is found most often and the circle indicates the local optimum with the highest posterior probability. Random Gaussian noise with standard deviation 0.5 is added to the x-values to aid the illustration.

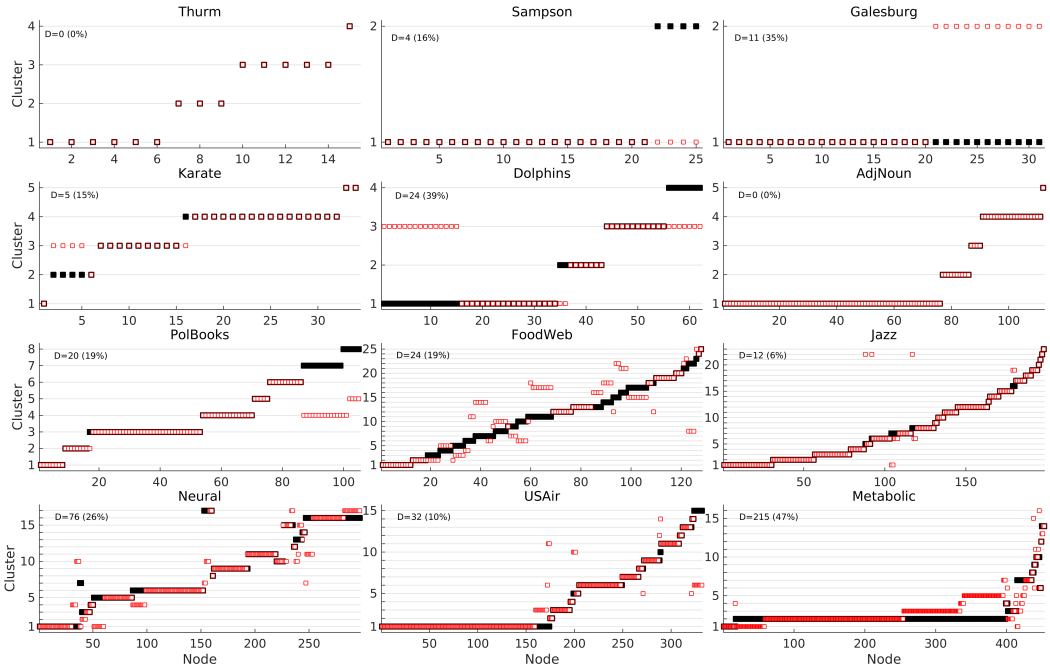


Figure 4: Comparison of the most often found (black) and the best (red) clustering (solutions marked in Fig. 3.) *D* denotes the smallest number (percentage) of nodes that must be reassigned to move between the two clusterings.

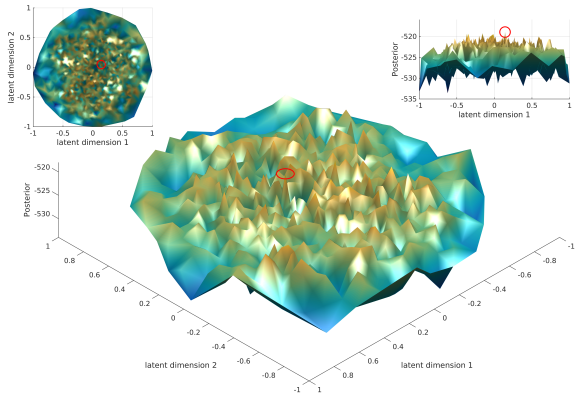


Figure 5: Latent embedding plot (inspired by [1]) of the posterior distribution of the Dolphins network, centered on the best local optimum. The figure was generated by finding all clusterings within a distance of 3 cluster reassignments from those visited by the MCMC procedure as well as their local optima. These solutions were mapped to a 2-dimensional latent space using curvilinear component analysis based on the minimum length sequence distance.

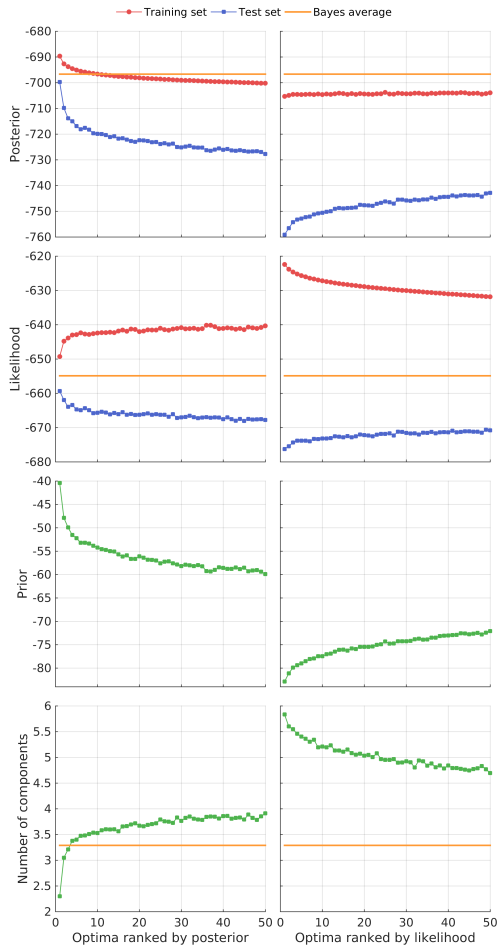


Figure 6: Predictive performance on held-out test data of the top 50 local optima averaged over ensemble of 500 networks with $N = 50$ nodes generated from the IRM. Local optima are ranked by posterior probability (top row) and likelihood (bottom row) on training data. Also shown is Bayes optimal predictive performance, the value of the prior, and the number of components.

References

[1] B. H. Good, Y.-A. de Montjoye, and A. Clauset, "The performance of modularity maximization in practical contexts," Phys. Rev. E 81, 046106, 2010.

[2] B. D. McKay and A. Piperno, Practical Graph Isomorphism, II, J. Symbolic Computation (2013) 60 94-112. http://dx.doi.org/10.1016/j.jsc.2013.09.003 Preprint version at arxiv.org.

[3] H. C. White, S. A. Boorman, and R. L. Breiger. "Social structure from multiple networks. i. blockmodels of roles and positions.," American journal of sociology, pages 730–780, 1976.

[4] The On-Line Encyclopedia of Integer Sequences, Sequence A000088, http://oeis.org/A000088

[5] The On-Line Encyclopedia of Integer Sequences, Sequence A000110, http://oeis.org/A000110

[6] P. L. Green, and K. Worden "Bayesian and Markov chain Monte Carlo methods for identifying nonlinear systems in the presence of uncertainty," *Phil. Trans. R. Soc. A* 373: 20140405.

[7] Stephen P. Brooks, and A. Gelman, "General Methods for Monitoring Convergence of Iterative Simulations," *Journal of Computational and Graphical Statistics*, vol. 7, no. 4, pp. 434–455, 1998

[8] S. P. Brooks, P. Giudici, and A. Philippe, "Nonparametric convergence assessment for MCMC model selection," *Journal of Computational and Graphical Statistics*, vol. 12, no. 1, pp. 1–22, 2003

[9] R. M. Neal, "Markov Chain Sampling Methods for Dirichlet Process Mixture Models," *Journal of Computational and Graphical Statistics*, vol. 9, no. 2, pp. 249-265, 2000

[10] T. S. Ferguson, "A Bayesian Analysis of Some Nonparametric Problems," *The Annals of Statistics*, vol. 1, no. 2, pp. 209-230, 1973

[11] C. E. Antoniak, "Mixtures of Dirichlet Processes with Applications to Bayesian Nonparametric Problems," *The Annals of Statistics*, vol. 2, no. 6, pp. 1152-1174, 1974

[12] Y. W Teh, "Dirichlet process," *Encyclopedia of machine learning*, pp. 1152-1174, Springer US, 2010

[13] M. N. Schmidt and M. Mørup, "Nonparametric Bayesian modeling of complex networks: An introduction," 2013.

[14] C. Kemp, J. B. Tenenbaum, T. L. Griffiths, T. Yamada, and N. Ueda, "Learning systems of concepts with an infinite relational model," *Cognitive Science*, vol. 21, no. 1, pp. 381, 2006.

[15] Z. Xu, V. Tresp, K. Yu, and H. Kriegel, "Infinite Hidden Relational Models," in *Proceedings of the 22nd International Conference on Uncertainity in Artificial Intelligence*, 2006.

[16] T. A. B. Snijders and K. Nowicki, "Estimation and prediction for stochastic blockmodels for graphs with latent block structure," *Journal of Classification*, vol. 14, no. 1, pp. 75–100, 1997.

[17] D. J. Aldous, "Exchangeability and related topics," Springer Berlin Heidelberg, 1985

[18] A. F. M. Smith and G. O. Roberts, "Bayesian Computation Via the Gibbs Sampler and Related Markov Chain Monte Carlo Methods," *Journal of the Royal Statistical Society* Series B (Methodological), vol. 55, no. 1, pp. 3–23, 1993.

[19] B. Larget, and D. L. Simon, "Markov chain Monte Carlo algorithms for the Bayesian analysis of phylogenetic trees," *Molecular Biology and Evolution* Series B (Methodological), vol. 16, no. 6, pp. 750–759, 1999.

[20] K. J. Albers, A. L. Moth, M. Mørup, and M. N. Schmidt, "Large scale inference in the Infinite Relational Model: Gibbs sampling is not enough," in *IEEE International Workshop on Machine Learning for Signal Processing, MLSP*, 2013.

[21] S. Jain and R. M. Neal, "A split-merge markov chain monte carlo procedure for the dirichlet process mixture model," *Journal of Computational and Graphical Statistics*, vol. 13, no. 1, 2004.

[22] A. J. Stam, "Generation of a random partition of a finite set by an urn model," *Journal of Combinatorial Theory, Series A*, vol. 35, no. 2, pp. 231–240, 1983

[23] D. E. Knuth, "The art of computer programming; Generating all combinations and partitions," vol. 4, fasc. 3, Addison-Wesley, 1998.

[24] C. L. DuBois, E. S. Spiro, Z. Almquist, M. S. Handcock, D. Hunter, C. T. Butts, S. M. Goodreau, and M. Morris, "2003 netdata: A Collection of Network Data," Network Data Repository, University of California, https://networkdata.ics.uci.edu/

[25] V. Batagelj and A. Mrvar, "Pajek datasets," 2006, http://vlado.fmf.uni-lj.si/pub/networks/data/

[26] Center for Computational Analysis of Social and Organizational Systems (CASOS) , Carnegie Mellon University, http://www.casos.cs.cmu.edu/

[27] L. L, L. Pan, Tao Zhou, Y. Zhang, and H. E. Stanley "Toward link predictability of complex networks," *PNAS*, vol. 112, no 8, pp. 2325-2330, 2015

[28] B. Thurman "In the office: Networks and coalitions," *Social Networks*, vol. 2, no 2, pp. 47-63, 1979

[29] S. F. Sampson, "A Novitiate in a Period of Change. An Experimental and Case Study of Social Relationships," *PhD thesis, Cornell University*, 1968

[30] J. S. Coleman, and E. Katz, "Medical Innovation," 1966. Data obtained from http://vlado.fmf.uni-lj.si/pub/networks/Data/esna/Galesburg2.htm

[31] D. Lusseau, K. Schneider, O. J. Boisseau, P. Haase, E. Slooten, and S. M. Dawson, "The bottlenose dolphin community of Doubtful Sound features a large proportion of long-lasting associations," *Behavioral Ecology and Sociobiology*, vol. 54, pp. 396-405, 2003

[32] W. W. Zachary, "An information flow model for conflict and fission in small groups," *Journal of Anthropological Research*, vol. 33, pp. 452-473, 1977

[33] M. E. J. Newman, "Finding community structure in networks using the eigenvectors of matrices," *Phys. Rev.*, E 74, 2006

[34] D. J. Watts, and S. H. Strogatz, "Collective dynamics of small-world networks," *Nature*, vol. 393, pp. 440-442, 1998

[35] J. Duch, A. Arenas, "Community detection in complex networks using extremal optimization," *Phy Rev*, E 72, 2005

[36] R. E. Ulanowicz, C. Bondavalli, and M. S. Egnotovich "Network analysis of trophic dynamics in South Florida ecosystems, FY 97: the Florida Bay ecosystem," Technical report, CBL 98-123, Chesapeake Biological Laboratory, Maryland (USA), 2005

[37] P. Gleiser, and L. Danon, "Community structure in Jazz," *Advances in complex systems*, vol. 6(4), pp. 565-573, 1998

Appendix D

Nonparametric Bayesian Clustering of Structural Whole Brain Connectivity in Full Resolution

Nonparametric Bayesian Clustering of Structural Whole Brain Connectivity in Full Resolution. Karen Sandø Ambrosen, Kristoffer Jon Albers, Tim B. Dyrby, Mikkel N. Schmidt, and Morten Mørup. In Pattern Recognition in NeuroImaging (PRNI), 2014.

Nonparametric Bayesian Clustering of Structural Whole Brain Connectivity in Full Image Resolution

Karen Sandø Ambrosen*†, Kristoffer Jon Albers*, Tim B. Dyrby†, Mikkel N. Schmidt*, and Morten Mørup*
*Department of Applied Mathematics and Computer Science, Technical University of Denmark, Lyngby, Denmark
†Danish Research Centre for Magnetic Resonance, Copenhagen University Hospital Hvidovre, Hvidovre, Denmark

***Abstract*—Diffusion magnetic resonance imaging enables measuring the structural connectivity of the human brain at a high spatial resolution. Local noisy connectivity estimates can be derived using tractography approaches and statistical models are necessary to quantify the brain's salient structural organization. However, statistically modeling these massive structural connectivity datasets is a computational challenging task. We develop a high-performance inference procedure for the infinite relational model (a prominent non-parametric Bayesian model for clustering networks into structurally similar groups) that defines structural units at the resolution of statistical support. We apply the model to a network of structural brain connectivity in full image resolution with more than one hundred thousand regions (voxels in the gray-white matter boundary) and around one hundred million connections. The derived clustering identifies in the order of one thousand salient structural units and we find that the identified units provide better predictive performance than predicting using the full graph or two commonly used atlases. Extracting structural units of brain connectivity at the full image resolution can aid in understanding the underlying connectivity patterns, and the proposed method for large scale data driven generation of structural units provides a promising framework that can exploit the increasing spatial resolution of neuro-imaging technologies.**

I. Introduction

Diffusion magnetic resonance imaging (dMRI) is an important non-invasive technique for studying the brain's structural organization. By tracking the diffusion of mainly water molecules that align with the orientation of the fibers in the brain, local estimates of fiber orientation can be obtained. These estimates are aggregated by tractography to derive maps of structural connectivity between cortical gray matter regions [5]. For the current dMRI technology these maps in full image resolution constitute complex networks of structural connectivity in the order of one hundred thousand regions and one hundred million links (see Fig. 1).

While the quantified fiber orientation within small regions of the brain as well as the subsequently derived local connectivity estimates are very noisy, these estimates can be aggregated to derive networks of whole brain connectivity within larger regions of structural units. These structural units have traditionally been based on automatic subdivision of the human brain into a fixed number of pre-specified neuroanatomical regions of interests (ROIs) [13], [7]. The Destrieux atlas [13], [8] currently has around 150 ROIs whereas the Desikan-Killiany atlas [7] has 68 ROIs. While these ROIs can be arbitrarily subdivided to provide additional regions [14] they are not explicitly based on the evidence obtained by the structural connectivity data and may therefore not optimally reflect the latent connectivity patterns of structural connectivity. Rather than fixing the structural units to a predefined atlas, we set out to learn the number of structural units and their spatial representations from the raw high resolution networks obtained

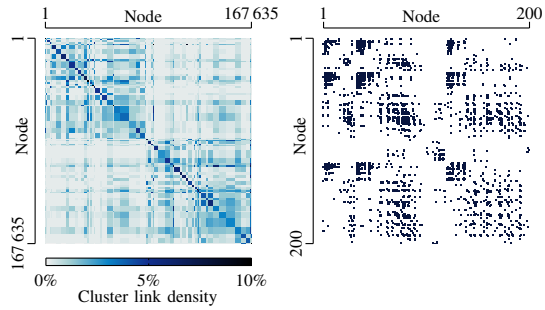


Fig. 1. Complex network of structural brain connectivity with 167 635 nodes and around one hundred million links obtained using 5000 streamlines per seed voxel. Left: Link density in each pair of the 68 regions of interest in the Desikan-Killiany atlas. Right: Links between the first 200 regions.

using tractography. To accomplish this we develop a large scale implementation of a prominent statistical network model, the infinite relational model (IRM) [17], [24]. The IRM is able to infer structurally consistent units at a resolution which is determined based on statistical evidence. While structural connectivity graphs have previously been clustered based on IRM [3] as well as other tools such as modularity [14], this is to the best of our knowledge the first attempt at modelling structural connectivity at the full image resolution of current dMRI technology.

This paper examines the capabilities of a large scale implementation of the IRM to identify structure in high-resolution structural brain connectivity graphs. We study to what extend we can perform inference on such large scale networks and whether it is feasible to reliably detect the structural units in a data driven manner using our implementation. In particular, we investigate: i) *What is the statistically salient resolution of structural connectivity graphs*, i.e., how many clusters are used to represent high resolution structural connectivity data? ii) *How reliable can these salient structures be detected*, i.e. how consistent are the structural units with respect to initialization and convergence of the sampler as well as the number of streamlines? iii) *Are the derived structural units better at predicting connectivity than existing atlases*, i.e. how well does the connectivity patterns derived from the structural units of one graph predict the connectivity of another graph obtained from another set of whole brain diffusion weighted images from the same subject?

II. Statistical Model and Inference

A. Infinite Relational Modelling

The Infinite Relational Model (IRM) [17], [24] is a non-parametric extension of the stochastic block model [19] in

which vertices in a graph are grouped into homogenous blocks according to their structural similarity. The IRM uses the Chinese Restaurant Process (CRP) [2], [20] as prior for the partitioning of vertices to groups thereby allowing for an arbitrary number of groups. The IRM is defined by the following generative process:

$$\boldsymbol{z} \sim \mathrm{CRP}(\alpha), \qquad \text{groups}, \tag{1}$$

$$\eta_{lm} \sim \mathrm{Beta}(\beta^+, \beta^-), \qquad \text{interactions}, \tag{2}$$

$$A_{ij} \sim \mathrm{Bernoulli}(\eta_{z_i z_j}), \qquad \text{links}, \tag{3}$$

where $\boldsymbol{z}$ is the group assignment, $\boldsymbol{\eta}$ is the probability of links between each pair of groups, and $\boldsymbol{A}$ is the adjacency matrix of the graph. As the beta prior on the elements of $\boldsymbol{\eta}$ is conjugate to the Bernoulli likelihood these parameters can be analytically integrated to form the joint distribution:

$$P(\boldsymbol{A}, \boldsymbol{z}|\alpha, \beta^+, \beta^-) = \int P(\boldsymbol{A}, \boldsymbol{z}, \boldsymbol{\eta}|\alpha, \beta^+, \beta^-) d\boldsymbol{\eta} \tag{4}$$

$$= \frac{\alpha^K \Gamma(\alpha) \prod_k \Gamma(n_k)}{\Gamma(J+\alpha)} \prod_{l \leq m} \frac{\mathrm{B}(N_{lm}^+ + \beta^+, N_{lm}^- + \beta^-)}{\mathrm{B}(\beta^+, \beta^-)},$$

where K is the number of groups, J is the number of vertices, n_k is the number of vertices assigned to the k'th group, N_{lm}^+ and N_{lm}^- are the number of links and non-links between group l and m, and $B(a,b) = \frac{\Gamma(a)\Gamma(b)}{\Gamma(a+b)}$ is the beta function.

B. Inference by Markov Chain Monte Carlo (MCMC)

To infer the posterior distribution, $P(\boldsymbol{z}|\boldsymbol{A}, \alpha, \beta^+, \beta^-)$, we use an MCMC procedure combining Gibbs and split-merge sampling [17]. In Gibbs sampling the posterior conditional distribution of placing one vertex at a time in any of the existing groups or in a new empty group is evaluated and the vertex is assigned according to this distribution. The probability of assigning a vertex i to group ℓ is given by:

$$P(z_i = \ell|\boldsymbol{A}, \boldsymbol{z}_{\setminus i}, h) = \frac{P(\boldsymbol{A}, \boldsymbol{z}_{\setminus i}, z_i = \ell|h)}{\sum_{\ell'=1}^{K+1} P(\boldsymbol{A}, \boldsymbol{z}_{\setminus i}, z_i = \ell'|h)}, \tag{5}$$

where $h = \{\beta^+, \beta^-, \alpha\}$ denotes the hyperparameters.

Rather than considering the assignment of a single vertex at a time split-merge sampling as presented in [15] attempts to merge or split existing clusters. Here, two vertices i and j are selected at random. If they are currently assigned to two different groups $z_i \neq z_j$, it is proposed to merge the two groups. Else it is proposed to split the single group in two. The procedure makes use of Gibbs sampling restricted to the nodes of the considered group(s) in order to define an intermediate launch state as well as to define the final split configuration and its transition probability $q(z|z^*)$. For a split configuration $q(z|z^*)$ is derived as the product of the individual transition probabilities of the vertices to move from the launch state to the final split configuration. As a merge transition is deterministic the transition from a split to a merge configuration has probability 1. Proposals are rejected or accepted according to the Metropolis-Hastings acceptance probability:

$$\alpha(z^*|z) = \min\left[1, \frac{P(\boldsymbol{A}, \boldsymbol{z}^*|\beta^+, \beta^-, \alpha) q(z|z^*)}{P(\boldsymbol{A}, \boldsymbol{z}|\beta^+, \beta^-, \alpha) q(z^*|z)}\right]. \tag{6}$$

C. Large scale computation

To get the computational performance necessary for the IRM to model structural connectivity in full image resolution we used a dedicated implementation optimized towards fully utilizing the memory structure and processor architecture of modern computers (see [1] for details). As the restricted Gibbs sweeps turns out to be the most computational demanding part of the split-merge sampling procedure, the performance of both sampling strategies benefits from most of the same optimizations. We store data in appropriate structures such that the sampling algorithms access data elements from sequential memory. In this way the access pattern takes advantage of the memory cache structure allowing for significantly faster memory accesses. To further speed up the Gibbs sampler we store and update the sufficient statistics, N^+ and N^-, instead of recalculating them in every Gibbs sweep. To ensure numeric stability within machine precision, the posterior in Eq. 5 is calculated in the log domain. The key operation then becomes calculating the logarithm of the beta function which relies on the gamma-function, $\Gamma(a)$, as:

$$\log B(a,b) = \log\Gamma(a) + \log\Gamma(b) - \log\Gamma(a+b) \tag{7}$$

As we only allow integer values for the hyperparameters, we use a lookup table of precalculated values for $\log\Gamma(a)$ which speeds up the evaluation of the posterior.

III. Data

To validate our proposed method, we used a dMRI data set previously described in [23], [22]. The data was collected at Danish Research Center for Magnetic Resonance and the study was approved by the local ethics committee. One healthy subject was scanned. The images were acquired on a Siemens VERIO 3T scanner using a 32-channel head coil. Two high resolution T1-weighted MRI images were acquired using a TR of 1,900 ms, TE of 2.32 ms, a FA of 9°, and 0.9mm^3 isotropic resolution. Two sets of whole brain diffusion weighted images (DWI) were acquired in 61 non-collinear directions with a b-value of b = 1500 s/mm^2, and ten non-diffusion weighted images (b = 0 s/mm^2). For this the twice refocused spin echo sequence with a TR of 11,440 ms and a TE of 89 ms. 61 axial slices with a resolution of 2.3 mm^3 isotropic voxels and Grappa = 2 were acquired [21]. A field map was acquired using a double gradient echo sequence with a TR of 479 ms, TE1 of 4.92 ms, TE2 of 7.38 ms, and a resolution of 3mm^3 isotropic voxels. The diffusion weighted images (DWI) were pre-processed using SPM8 (www.fil.ion.ucl.ac.uk/spm). To reduce motion artifacts and eddy current induced distortions an affine transformation between the DWIs based on normalized mutual information was applied. The voxel displacement map (VDM) was calculated based on the field map resliced to DWI resolution using the field map toolbox of SPM8 [16]. The VDM was applied to minimize geometric distortions due to susceptibility artifacts. Finally the DWIs were aligned and resliced with affine matrix to a T1 weighted MRI using 7th degree B-spline interpolation [10]. The 61 non-collinear diffusion weighting gradient directions were updated using the same rotations and transformations as the resliced images [18]. Segmentation of the white and gray matter was performed based on the high resolution structural T1w images using Freesurfer (surfer.nmr.mgh.harvard.edu) [6], [12], [11]. The Freesurfer reconstruction outputs, among others, the white matter segmentation and the gray-white matter boundary. The

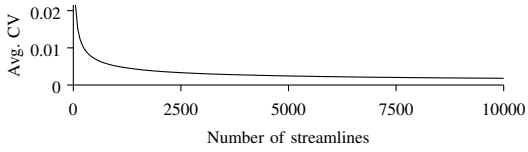


Fig. 2. Average voxel-wise coefficient of variation (CV) as function of number of streamlines in the tractography. The CV and the SNR are based on tractography results repeated five times for each number of streamlines.

gray-white matter boundary for both hemispheres was converted to volumes and transformed from Freesurfer conformed space to native space. Likewise, the white matter segmentation from the Freesurfer reconstruction was transformed to native space. The diffusion parameters were estimated using FSL's BedpostX and probabilistic tractography was performed using FSL's Probtrackx2 with the omatrix3 option [4]. The transformed white matter volume was used as seed in the tractography and the transformed cortex labels as both target and stop mask in the tractography. For all other options the default settings were used. The cortex to cortex connectivity graph were output from FSL's probtrackx2 using the omatrix3 option. We obtained four 167,635×167,635 connectivity graphs (i.e., scan and rescan for 1000 and 5000 streamlines per seed voxel). Each link in the graphs took on the value of the number of streamlines connecting the two voxels in the target mask (gray/white matter boundary). The graphs were symmetrized and binarized (i.e., for each graph the graph and its transpose were added together and entries that were subsequently above zero set to one).

IV. Experiments and Results

A. Number of streamlines

To ensure that the network obtained by tractography is robust, probabilistic tractography was performed with different number of streamlines: Between 50 and 10,000 streamlines per seed voxel were used. Each number of streamlines was run five times. The voxel-wise coefficient of variation (CV) between voxels within the seed mask in the images with equal number of streamlines was calculated as $\mathrm{CV} = \frac{\sigma}{\mu}$, where σ is the standard deviation and μ is the mean. The average CV across all voxels was calculated [9] and is shown as function of number of streamlines in Fig. 2. The number of streamlines used in the subsequent experiments was selected on the basis of the average CV: As the average CV seems to have reached a stable level when using 1000 streamlines, and definitely when using 5000 streamlines (Fig. 2) we compare these two values.

B. Model parameters, inference, and convergence

For each network we performed 10 separate runs, all with the hyper parameters $\beta^+ = \beta^- = 1$ and $\alpha = \lfloor \log(J) \rfloor$, where J is the total number of nodes. For each run, we performed 100 iterations of the following sampling procedure: Each iteration began with a complete Gibbs sweep over all nodes. It was then followed by the same number of split-merge operations as the current number of clusters. In each split-merge operation we performed 10 restricted Gibbs sweeps. Each of these iterations took several hours to compute. Fig. 3 shows the logarithm of the joint distribution for the different runs. It is clear that the MCMC sampler does not converge (which was also not to be expected [1]), but even when the sampler does

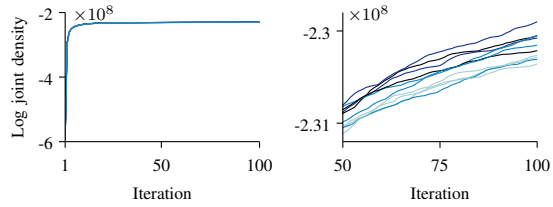


Fig. 3. Logarithm of the joint distribution for the MCMC inference procedure for the network based on 5000 streamlines. A zoom of the last 50 iterations is shown to the right.

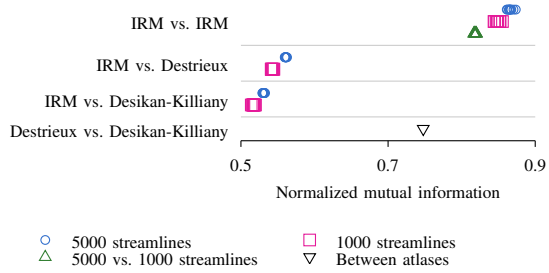


Fig. 4. Normalized Mutual Information (NMI) between 10 independent runs of the IRM and two atlases for network based on 1000 and 5000 streamlines.

not converge, the inferred grouping captures suboptimal but relevant structures in the network. In the following we used the inferred group structure after the last MCMC iteration.

C. Comparison and stability of estimated group structure

To compare the unsupervised groupings found by IRM with the groupings provided by the two atlases, we use the normalized mutual information (NMI) as a measure of similarity between 0 and 1. For two groupings $\boldsymbol{z}$ and $\boldsymbol{z}'$, we use: $\mathrm{NMI}(\boldsymbol{z}, \boldsymbol{z}') = \frac{2 \cdot \mathrm{I}(\boldsymbol{z}, \boldsymbol{z}')}{\mathrm{H}(\boldsymbol{z}) + \mathrm{H}(\boldsymbol{z}')}$ where $\mathrm{I}(\boldsymbol{z}, \boldsymbol{z}')$ is the mutual information between the groupings and $\mathrm{H}(\boldsymbol{z})$ is the entropy of $\boldsymbol{z}$. Fig. 4 shows NMI between all runs as well as between the runs and the two atlases. It is evident that the inferred groupings are very similar in the 10 runs as evidenced by the relatively high NMI, both within and between the networks based on 1000 and 5000 streamlines, respectively. Also, the inferred grouping is somewhat similar to the two atlases with an NMI score around 0.5-0.6.

D. Predictive performance

To assess how well the inferred structure fits the data, we use a second structural connectivity network based on a rescan of the same subject. Since any differences between the two scans are due to noise in the processes of generating the network, measuring how well we can predict the links in the second graph can be used to quantify the utility of the inferred structural units. To measure the predictive performance we use the area under the receiver operating characteristic curve (AUC) which allows us to compare predictions from the IRM model with predictions made using other existing atlases or predicting directly from the raw network data. The results in Fig. 5 show that the IRM model outperforms predictions from the raw graph as well as both the Desikan-Killiany and Destrieux atlases both for networks based on 1000 and 5000

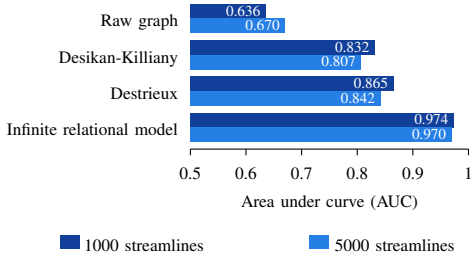


Fig. 5. Performance as measured by the area under the receiver operating characteristic curve (AUC) when predicting links in a network of structural connectivity based on data from a second scan of the same subject.

streamlines. However, when inspecting the extracted structural units (not shown) they were more diffuse compared to the atlases which may hamper their interpretation. This may be attributed both to the lack of convergence as well as lack of spatial constraints in the modeling.

V. Conclusion

When analyzing whole brain structural connectivity in full image resolution in the order of one thousand salient structural units were identified by our large scale implementation of the infinite relational model. The network based on 5000 streamlines had more structural units compared to the network based on 1000 streamlines. However, the estimated group structures were quite similar as quantified by NMI. Although the MCMC sampler did not reach convergence the identified groups were fairly robust to initialization while having some similarity to the Destrieux and Desikan-Killiany atlases. Notably, the extracted structural units provided significantly better predictive performances than predicting using the structural connectivity graph itself or the two considered atlases.

The present paper is to the best of our knowledge the first attempt at clustering structural connectivity in full resolution and provides a promising tool for a more detailed account of structural connectivity in general. In future work the influence of image resolution and choice of hyper-parameters should be investigated as should better sampling strategies.

Acknowledgment

This project was funded by the Lundbeck Foundation. We thank Andreas Leon Aagaard Moth for help with the large scale implementation.

References

[1] Kristoffer Jon Albers, Andreas Leon Aagard Moth, Morten Mørup, and Mikkel N. Schmidt. Large scale inference in the infinite relational model: Gibbs sampling is not enough. In *Machine Learning for Signal Processing (MLSP), 2013 IEEE International Workshop on*, pages 1–6. IEEE, 2013.

[2] D. Aldous. Exchangeability and related topics. *École d'Été de Probabilités de Saint-Flour XIII1983*, pages 1–198, 1985.

[3] Karen S Ambrosen, Tue Herlau, Tim Dyrby, Mikkel N Schmidt, and Morten Mørup. Comparing structural brain connectivity by the infinite relational model. In *Pattern Recognition in Neuroimaging (PRNI), 2013 International Workshop on*, pages 50–53. IEEE, 2013.

[4] TEJ Behrens, H Johansen Berg, Saad Jbabdi, MFS Rushworth, and MW Woolrich. Probabilistic diffusion tractography with multiple fibre orientations: What can we gain? *Neuroimage*, 34(1):144–155, 2007.

[5] TEJ Behrens, MW Woolrich, M Jenkinson, H Johansen-Berg, RG Nunes, S Clare, PM Matthews, JM Brady, and SM Smith. Characterization and propagation of uncertainty in diffusion-weighted mr imaging. *Magnetic resonance in medicine*, 50(5):1077–1088, 2003.

[6] Anders M Dale, Bruce Fischl, and Martin I Sereno. Cortical surface-based analysis: I. segmentation and surface reconstruction. *Neuroimage*, 9(2):179–194, 1999.

[7] Rahul S Desikan, Florent Ségonne, Bruce Fischl, Brian T Quinn, Bradford C Dickerson, Deborah Blacker, Randy L Buckner, Anders M Dale, R Paul Maguire, Bradley T Hyman, et al. An automated labeling system for subdividing the human cerebral cortex on mri scans into gyral based regions of interest. *Neuroimage*, 31(3):968–980, 2006.

[8] Christophe Destrieux, Bruce Fischl, Anders Dale, and Eric Halgren. Automatic parcellation of human cortical gyri and sulci using standard anatomical nomenclature. *Neuroimage*, 53(1):1–15, 2010.

[9] Annette J Dobson. *An introduction to generalized linear models*. CRC press, 2001.

[10] TB Dyrby, HM Lundell, MG Liptrot, MW Burke, M Ptito, and HR Siebner. Interpolation of dwi prior to dti reconstruction, and its validation. In *Proc. Intl. Soc. Mag. Reson. Med*, volume 19, 2011.

[11] Bruce Fischl, David H Salat, Evelina Busa, Marilyn Albert, Megan Dieterich, Christian Haselgrove, Andre van der Kouwe, Ron Killiany, David Kennedy, Shuna Klaveness, et al. Whole brain segmentation: automated labeling of neuroanatomical structures in the human brain. *Neuron*, 33(3):341–355, 2002.

[12] Bruce Fischl, Martin I Sereno, and Anders M Dale. Cortical surface-based analysis: Ii: Inflation, flattening, and a surface-based coordinate system. *Neuroimage*, 9(2):195–207, 1999.

[13] Bruce Fischl, André van der Kouwe, Christophe Destrieux, Eric Halgren, Florent Ségonne, David H Salat, Evelina Busa, Larry J Seidman, Jill Goldstein, David Kennedy, et al. Automatically parcellating the human cerebral cortex. *Cerebral cortex*, 14(1):11–22, 2004.

[14] Patric Hagmann, Leila Cammoun, Xavier Gigandet, Reto Meuli, Christopher J Honey, Van J Wedeen, and Olaf Sporns. Mapping the structural core of human cerebral cortex. *PLoS biology*, 6(7):e159, 2008.

[15] S. Jain and Radford M Neal. A split-merge markov chain monte carlo procedure for the dirichlet process mixture model. *Journal of Computational and Graphical Statistics*, 13(1), 2004.

[16] Peter Jezzard and Robert S Balaban. Correction for geometric distortion in echo planar images from b0 field variations. *Magnetic resonance in medicine*, 34(1):65–73, 1995.

[17] C. Kemp, J. B. Tenenbaum, T. L. Griffiths, T. Yamada, and N. Ueda. Learning systems of concepts with an infinite relational model. In *AAAI*, volume 3, page 5, 2006.

[18] Alexander Leemans and Derek K Jones. The b-matrix must be rotated when correcting for subject motion in dti data. *Magnetic Resonance in Medicine*, 61(6):1336–1349, 2009.

[19] Krzysztof Nowicki and Tom A B Snijders. Estimation and prediction for stochastic blockstructures. *Journal of the American Statistical Association*, 96(455):1077–1087, 2001.

[20] J. Pitman et al. Combinatorial stochastic processes. Technical report, Springer, 2002.

[21] TG Reese, O Heid, RM Weisskoff, and VJ Wedeen. Reduction of eddy-current-induced distortion in diffusion mri using a twice-refocused spin echo. *Magnetic Resonance in Medicine*, 49(1):177–182, 2003.

[22] Nina Linde Reislev, Ron Kupers, Hartwig R. Siebner, Maurice Ptito, and Tim B. Dyrby. Blindness differentially affects the integrity of the dorsal and ventral visual streams, 2013.

[23] Nina Linde Reislev, Maurice Ptito, Ron Kupers, Hartwig R. Siebner, and Tim B Dyrby. Alterations of the inferior longitudinal fasciculus in congenital and late blindness, 2012.

[24] Z. Xu, V. Tresp, K. Yu, and H.P. Kriegel. Learning infinite hidden relational models. *Uncertainty in Artificial Intelligence (UAI2006)*, 2006.

Appendix E

Predictive Validation of Human Brain Parcellation

Predictive Validation of Human Brain Parcellations. Karen Sandø Ambrosen, Kristoffer Jon Albers, Matthew G. Liptrot, Tim B. Dyrby, Mikkel N. Schmidt, and Morten Mørup. (2017, under review).

Predictive Validation of Human Brain Parcellation

Karen S. Ambrosen · Kristoffer J. Albers · Matthew G. Liptrot · Tim B. Dyrby · Mikkel N. Schmidt · Morten Mørup

Abstract The organization of the human brain remains elusive, yet is of great importance to the mechanisms of integrative brain function. At the macroscale, its structural and functional interpretation is conventionally assessed at the level of cortical units. However, the definition and validation of such cortical parcellations are problematic due to the absence of a true gold standard. We propose a framework for quantitative validation of brain parcellations via statistical prediction on independent brain connectivity data. We assess the pertinence of three existing parcellations to account for structural connectivity (SC) data, and compare them to data-driven parcellations optimized for SC and random parcellations. We find that all three atlases perform better than random, and that one, a recently proposed multi-modal atlas, provides superior characterisation of SC compared to those based solely upon surface morphology. Our analysis further suggests that the SC data is better characterized by the inclusion of more parcels than those contained in the considered atlases.

Keywords Brain parcellation · Diffusion magnetic resonance imaging (dMRI) · Whole brain structural connectivity · Human connectome · Link prediction

1 Introduction

The vast complexity of the human brain [Braitenberg and Schüz, 1991, Murre and Sturdy, 1995] and the incomplete and noisy measurements available through neuroimaging modalities require a pragmatic approach to the analysis of the human connectome [Sporns et al., 2005, Hagmann, 2005]. Segregation into anatomical or functional units provides interpretable and noise reduced network nodes whose inter-connections approximate the brain's organizational structure [Bullmore and Sporns, 2009, Sporns, 2012]. Much research is underway to delineate the structural and functional organization of the human brain [Smith, 2013, Van Essen et al., 2013b] but it remains unclear which parcellation best accounts for such organization and how this is quantified.

To provide a sound basis for analysis, the nodes provided by a given parcellation method must be robust across a population, and fully represent their local infrastructure, microscopical properties and connectional "fingerprint" — their unique pattern of inputs and outputs [Passingham et al., 2002]. For example, when defining cortical regions at the macroscale it has been suggested that specific functions of the areas, such as connectivity, reproducibility, convergence, multimodality, evolutionary coherence, and inter-subject variability, should all be taken into account [Amunts and Zilles, 2015]. However, there still remains a lack of gold standard evaluation strategies against which any particular parcellation can be tested.

The exact method of parcellation employed depends upon the application. Hence a wide variety of parcellation schemes are currently available, including cortical surface morphology (the Desikan-Killiany atlas [Desikan et al., 2006], the Destrieux atlas [Fischl et al., 2004]), functional activation (the AAL atlas [Tzourio-Mazoyer et al., 2002, Gong et al., 2009]), parcellations derived from structural connectivity (SC) [Ambrosen et al., 2014, Parisot et al., 2016, Baldassano et al., 2015], and combinations thereof including Brainnetome [Fan et al., 2016] (cortical surface morphology and SC), and HCP_MMP1.0 [Glasser et al., 2016] (function- and structure-related features).

Karen S. Ambrosen (✉) · Kristoffer J. Albers · Matthew G. Liptrot · Tim B. Dyrby · Mikkel N. Schmidt · Morten Mørup
Department of Applied Mathematics and Computer Science, Technical University of Denmark, Richard Petersens Plads, Building 324, DK-2800 Kgs. Lyngby, Denmark
E-mail: kmsa@dtu.dk

Karen S. Ambrosen · Tim B. Dyrby
Danish Research Centre for Magnetic Resonance, Centre for Functional and Diagnostic Imaging and Research, Copenhagen University Hospital Hvidovre, Kettegaard All 30, DK-2650 Hvidovre, Denmark

The various parcellation schemes exhibit considerable differences, e.g. number of parcels and parcel border locations, and no single parcellation appears to be universally accepted. The situation is exacerbated by studies showing that subsequent graph measures are sensitive to the chosen parcellation scheme, both for structural [Hagmann et al., 2008] and functional [Zalesky et al., 2010, Fornito et al., 2010] analyses.

The differences in both the size, extent and downstream effects of a parcellation illustrate that it is important to validate its relevance to the application in question. Whilst reliability is often purported as a proxy for validation [Thirion et al., 2014, Fan et al., 2016, Glasser et al., 2016], it is not sufficient because a method can be arbitrarily reliable yet poorly account for brain organization. In contrast, the use of data on brain organization that is independent of how a parcellation is derived can in principle permit validation of a parcellation. To achieve this, we herein describe how parcellations can be validated using statistical prediction based upon independent brain connectivity data.

Our statistical prediction framework poses quantification of parcellation quality as a link-prediction problem [Liben-Nowell and Kleinberg, 2007, Clauset et al., 2008, Andersen et al., 2014, Ambrosen et al., 2014]. A parcellation is thereby assessed by its ability to characterize brain connectivity data from an independent modality. In particular, the approach quantifies how well network structure is preserved in the independent modality by the network organization induced by the parcellation. Herein, we have used independent high-resolution SC data from the Human Connectome Project (HCP) [Van Essen et al., 2012, Moeller et al., 2010, Feinberg et al., 2010, Setsompop et al., 2012, Xu et al., 2012] to validate three (non-SC derived) atlases: Desikan-Killiany (68 parcels) [Desikan et al., 2006], Destrieux (148 parcels) [Fischl et al., 2004] and Human Connectome Project multi-modality parcellation (HCP_MMP1.0, 360 parcels) [Glasser et al., 2016]. Whereas the first two are based upon surface morphology, the latter is a multi-modality atlas which includes fMRI (both resting-state and task-based), cortical thickness and myelin mapping. We contrast the predictive performance of these atlases to SC-informed parcellations, as well as to spatially-homogeneous random parcellations.

2 Materials and Methods

2.1 Diffusion imaging, tractography and construction of connectivity graphs

The MRI data used in the preparation of this work were obtained from the MGH-USC Human Connectome Project (HCP) database (https://ida.loni.usc.edu/login.jsp) in the "500 subjects" release. Acquisition parameters are described in full for dMRI in [Moeller et al., 2010, Feinberg et al., 2010, Setsompop et al., 2012, Xu et al., 2012, Sotiropoulos et al., 2013], and for the structural scans in [Milchenko and Marcus, 2013], and are listed in brief here. The dMRI was acquired with a multiband factor of 3, covering 270 directions distributed over 3 diffusion shells of b-values 1000, 2000 and 3000 s/mm^2, plus 18 $b = 0$ (non-diffusion weighted) scans. The nominal voxel size was $1.25mm$ isotropic. Both T_1-weighted and T_2-weighted structural scans at $0.7mm$ isotropic resolution were also acquired.

All pre-processing of the data, including correction for sequence-dependent artefacts such as eddy-current distortion, was performed by the "minimal preprocessing pipeline" provided by the HCP project [Glasser et al., 2013]. This included the generation of native pial and white-matter surfaces, and their coregistration to a standard vertex mesh. This provides a one-to-one correspondance between the surface vertices of every subject, and hence permits vertex-wise analysis of tractography results across the HCP population.

Tractography was performed using a GPU implementation of FSL's BedpostX [Hernández et al., 2013] and ProbtrackX2 [Jenkinson et al., 2012, Behrens et al., 2003b, Behrens et al., 2007]. BedpostX parameters included a specification of up to 3 fibres per voxel, and a deconvolution model using zeppelins [Behrens et al., 2003b, Behrens et al., 2007]. ProbtrackX2 was run in "matrix3" mode, with all voxels in the white matter (as specified by a structural imaging mask) as seed points. The GM-WM surface boundary and all subcortical grey-matter voxels were specified as tractography target masks. Streamlines were kept and entered into the resultant connectivity matrix if they succeeded in traversing opposite directions from a seed voxel and reaching two different target surface vertices or subcortical voxels. One thousand streamlines were generated from every seed voxel. The result of tractography is therefore a symmetric connectivity matrix of size $[(\textit{number of target surface vertices}) + (\textit{number of target subcortical voxels})]^2$. In this study, only the surface vertices are analysed. Thus, SC graphs ($n = 59,412$ vertices) covering the cerebral cortex of both hemispheres were generated for 26 subjects using data from the Human Connectome Project (HCP) [Van Essen et al., 2012, Moeller et al., 2010, Feinberg et al., 2010, Setsompop et al., 2012, Xu et al., 2012]. Each subject's SC graph was binarised by thresholding at a connectivity strength of 200 streamlines (see Online Resource 1 regarding the choice of threshold level). The structural connectivity graph of subject s can thus be represented by symmetric binary $J \times J$ adjacency matrix $\boldsymbol{A}^s$ such that $A^s_{ij} = 1$ and $A^s_{ij} = 0$ respectively denotes the existence or absence of a path from the tractography in either direction between i and j.

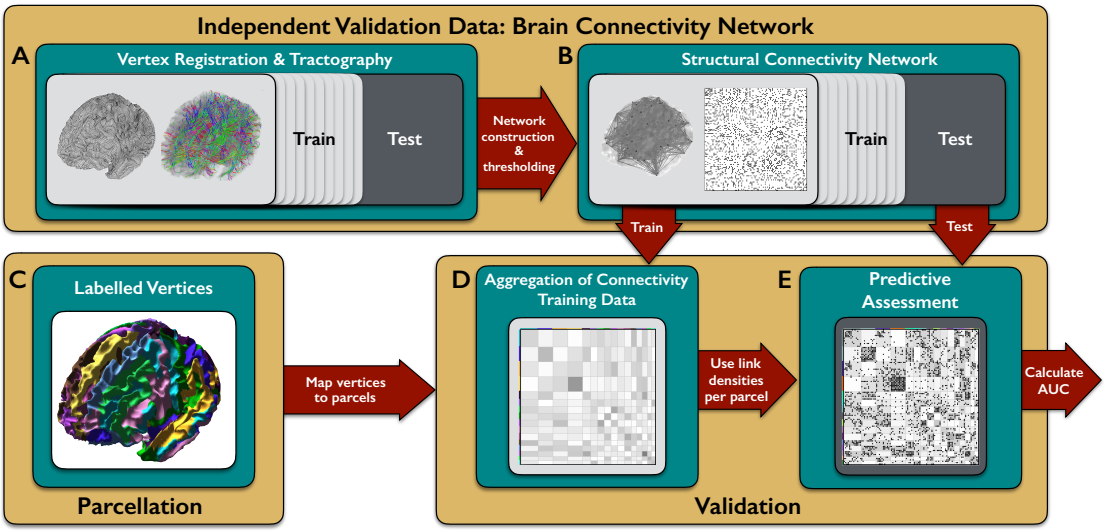


Fig. 1: The predictive validation framework using SC data. A) The native surfaces of all subjects are co-registered to a standard vertex mesh to obtain one-to-one correspondance between the surface vertices of every subject [Glasser et al., 2013]. Tractography is performed between all vertices of the surface by initialising 1000 streamlines in all white-matter voxels resulting in a weighted symmetric SC network for each subject. B) The networks are thresholded to obtain binary links of SC (connections in left panel, dots in right panel). C) The considered parcellation. D) The training networks are permuted according to the parcellation $\boldsymbol{z}$ in question and the link densities ρ_{lm} between and within parcels calculated by aggregating all the training networks. E) The predictive performance is assessed by calculating the area under the curve (AUC) of the receiver operator characteristics using the scores obtained from the training link-densities (grey background) to predict the links of the test network (overlaid dots).

2.2 The predictive validation framework

Figure 1 outlines the proposed predictive procedure. Input to the procedure is a parcellation $\boldsymbol{z}$ ($z_i = m$ indicates that node i belongs to parcel number m) and the SC networks of the training and test population considered. The average density ρ_{lm} of links for the training graphs is computed between each pair of parcels ($l \neq m$) and within each parcel ($l = m$), thereby representing the SC data in terms of the aggregated average connectivities between parcels based on the training population. These average connectivites are then used to predict the SC of the test graphs, by predicting a link between node i and j according to the score

$$s_{ij}^{\text{Parcellation}} = \rho_{z_i z_j} = \frac{N^+_{z_i z_j}}{N^+_{z_i z_j} + N^-_{z_i z_j}} \text{ for all } i > j, \qquad (1)$$

where $N^+_{lm} = \sum_s \sum_{i>j} A^s_{ij} \delta_{z_i=l} \delta_{z_j=m}$ and $N^-_{lm} = \sum_s \sum_{i>j} (1 - A^s_{ij}) \delta_{z_i=l} \delta_{z_j=m}$ respectively are the number of (aggregated) links and non-links in the training networks between nodes in cluster z_i and nodes in cluster z_j. Notably, this score corresponds to the maximum likelihood estimate of the connection probability $\eta_{z_i z_j}$ assuming a stochastic block model, see also Equation S14.

We evaluate the ability of these scores to predict links in holdout data in order to quantify how well structure is accounted for in the test graphs. A common procedure to quantify predictive performance when predicting links in networks is the area under curve (AUC) of the receiver operator characteristics (ROC) curve [Clauset et al., 2008, Miller et al., 2009]. By using the AUC score it is possible to compare predictions made using different parcellations as well as non-parametric link prediction measures [Liben-Nowell and Kleinberg, 2007, Clauset et al., 2008]. Links and non-links are scored using a given modeling approach, and the AUC then quantifies how well the two classes of links and non-links are separated according to this score, where an AUC score of 0.5 indicates that the scoring procedure is no better than chance whereas an AUC score of 1 indicates that a threshold value of the scoring procedure exists which provides a perfect separation of links from non-links. To provide values of reference for the scale of the AUC scores, we estimate upper- and lower-bounds on the predictive performance.

Upper bounds were estimated using data-driven SC parcellations, based on the stochastic block model (SBM) [White et al., 1976, Holland et al., 1983, Nowicki and Snijders, 2001, Schmidt and Mørup, 2013, Andersen et al., 2014, Ambrosen et al., 2014] as well as Ward clustering [Ward Jr, 1963] as proposed in [Eickhoff et al., 2011, Thirion et al., 2014, Baldassano et al., 2015]. These

methods cluster nodes into homogeneous parcels according to their structural similarity and are derived to optimally account for the SC profile of the whole brain. They thereby provide an estimate of the upper bound for the predictive performance that can be obtained on SC data using a parcellation.

In the SBM, nodes are partitioned into a given finite number of clusters K based on the Dirichlet distribution, allowing for flexible cluster sizes. The probability of links in the graph are generated according to a Bernoulli distribution, depending only upon the probability of observing links between clusters, which in turn follows a Beta distribution. The generative model is:

$$\begin{aligned}
\text{Links between nodes } A^s_{ij} &\sim Bernoulli(\eta_{z_i z_j}),\ \forall s, i > j \\
\text{Link densities between clusters } \eta_{lm} &\sim Beta(\beta^+, \beta^-),\ \forall l \geq m \\
\text{Clustering } z_i &\sim Categorical(\boldsymbol{\pi}),\ \forall i \\
\boldsymbol{\pi} &\sim Dirichlet(\boldsymbol{\alpha})
\end{aligned}$$

To solve the clustering problem we seek a common partition $\boldsymbol{z}$ of all subject's graphs, $\boldsymbol{A}^1, \ldots, \boldsymbol{A}^S$ into K clusters of nodes that exhibit similar structural connectivity patterns. For a particular data set the model parameters are inferred using a sequence of independent Markov Chain Monte Carlo methods to sample from the posterior distribution (see Online Resource 1 for details about the sampling procedure).

Ward clustering [Ward Jr, 1963] is initialised by assigning all nodes to their own cluster. In each step the two clusters that, when merged, produce the least increase of the objective function are combined together as a new cluster. The procedure terminates when all nodes are in the same cluster. We used the implementation provided by [Baldassano et al., 2015] using the squared dissimilarity measure $\mathbf{W}^2$ given by

$$W_{ij} = \sqrt{\sum_{a \neq i,j} (\tilde{A}_{ia} - \tilde{A}_{ja})^2 + \sum_{a \neq i,j} (\tilde{A}_{ai} - \tilde{A}_{aj})^2} \tag{2}$$

$$= \sqrt{2 \sum_{a \neq i,j} (\tilde{A}_{ia} - \tilde{A}_{ja})^2}, \tag{3}$$

where $\tilde{A}_{ij} = \sum_s A^s_{ij}$ and last equality follows for undirected graphs. The procedure additionally imposes the constraint that only clusters that are spatially adjacent can be merged.

To provide an estimate of the lower bound of predictive performance, we generated a parcellation based on k-means clustering [MacQueen et al., 1967] defined to group nodes based solely upon their Cartesian coordinates thereby forming spatially homogeneous random parcels uninformed by anatomy and SC. Given the vertices of the average surface $(v_1, v_2, ..., v_n)$, where each observation is a 3-dimensional real vector containing the (x,y,z)-coordinates of a vertex, k-means clustering partitions the n observations into k spatially-homogeneous clusters $C = C_1, C_2, ..., C_k$ in which each observation belongs to the cluster with the nearest mean. The k-means clustering iteratively assigns observations to clusters and updates the cluster means, $\boldsymbol{\mu}_i$, by minimizing the within-cluster sum of squares.

$$\arg\min_{\mathbf{C}} \sum_{i=1}^{k} \sum_{\mathbf{x} \in C_i} \|\mathbf{x} - \boldsymbol{\mu}_i\|^2. \tag{4}$$

Finally, to ascertain that no substantial information about SC is lost by representing SC data in terms of parcels, we contrast the performance of parcellations to conventional non-parametric link-prediction methods [Liben-Nowell and Kleinberg, 2007]. Let $d_i = \sum_j A^s_{ij}$ be the degree of node i. We use the following well-established measures to score for the existence of a link between node i and j [Liben-Nowell and Kleinberg, 2007, Clauset et al., 2008]:

$$s_{ij}^{\text{Common Neighbor}} = \sum_t A^s_{it} A^s_{jt} \tag{5}$$

$$s_{ij}^{\text{Jaccard}} = \frac{\sum_t A^s_{it} A^s_{jt}}{J - \sum_t (1 - A^s_{it})(1 - A^s_{jt})} \tag{6}$$

$$s_{ij}^{\text{Adamic/Adar}} = \sum_t \frac{A^s_{it} A^s_{jt}}{\log(d_t)} \tag{7}$$

$$s_{ij}^{\text{Preferential Attachment}} = d_i d_j \tag{8}$$

$$s_{ij}^{\text{ShortestPath}} = \frac{1}{\text{ShortestPath}(\boldsymbol{A}^s, i, j)}, \tag{9}$$

where $\text{ShortestPath}(\boldsymbol{A}^s, i, j)$ gives the shortest path in the structural connectivity graph $\boldsymbol{A}^s$. The above scores are averaged over the S training subjects and evaluated on holdout test data.

2.3 Considered Parcellations

We apply the predictive validation framework on the following three prominent non-SC based parcellations; Desikan-Killiany [Desikan et al., 2006], Destrieux [Destrieux et al., 2010], and HCP_MMP1.0 [Glasser et al., 2016].

2.3.1 The Desikan-Killiany atlas

The Desikan-Killiany atlas [Desikan et al., 2006] is based on a dataset of 40 MRI scans from a variety of subjects including young, middle-aged and elderly controls, as well as patients with Alzheimer's disease. Full details are available elsewhere ([Desikan et al., 2006]), but are repeated here in brief for completeness. A total of 34 cortical regions were manually identified in each hemisphere on volumetric T1-weighted MRI images using a 'sulcal' approach (manually tracing from the depth of one sulcus to another, thus incorporating the gyrus within) to define most structures, guided by standard neuroanatomical conventions based on brain atlases, modifications to previous published definitions and expert knowledge.

The volumetric ROIs were transposed onto the inflated cortical surface of each reconstructed brain and the final anatomic labels were generated using anatomic information regarding local curvature (e.g. the presence of sulci).

A cortical atlas was generated using a registration procedure that aligns the cortical folding patterns and probabilistically assigns a neuroanatomical region to every point in the cortical surface. This was done by generating a spherical representation of each brain by minimizing the metric distortion between the cortical and the spherical representations. The spherical surfaces were registered together. This established a spherical surface-based coordinate system that was adapted to the folding pattern of each individual subject, thus allowing for increased precision in registering anatomic features of the human brain across subjects. A spherical statistical atlas was used to label the cortical surfaces into neuroanatomical regions of interest.

2.3.2 The Destrieux atlas

The Destrieux atlas [Destrieux et al., 2010] is based on a parcellation scheme that first divided the cortex into gyral and sulcal regions, the limit between both being given by the values of local mean curvature or average convexity of the surfaces. Full details are available in [Destrieux et al., 2010], but are repeated here in brief for completeness. A gyrus was defined to be only the portion of the cortex that was visible on the pial view, whereas the remaining, hidden cortex (banks of sulci) were defined as belonging to a sulcus. For a few large structures, an additional sub-parcellation was used based on estimated cytoarchitectonic and functional criteria and some parcellations that were very small or very variable were grouped with a larger neighboring parcellation unit. Finally, each hemisphere was segmented into 74 different sulco-gyral cortical units.

A set of 12 subjects was used to develop and test the anatomical rules which labeled every point of the cerebral cortex, while another dataset of 12 subjects was used to train the automated labeling software.

The probability of a label at a certain vertex is based on a number of pieces of information, including the curvature and average convexity of the cortical surface, prior labeling probability for that vertex, as well as the labels of vertices in a local neighborhood.

2.3.3 The Human Connectome Project Multi-model Parcellation (HCP_MMP1.0) atlas

This recently released atlas [Glasser et al., 2016] comprises 180 parcels per hemisphere, and was generated using a novel combination of machine-learning and interactive editing by neuroanatomists. Using a combination of modalities, including fMRI, myelin maps and structural imaging, 210 subjects were aligned using an areal-matching algorithm, and subsequently the surface gradients of the different modalities were used to propose parcel borders. These were then edited and documented by neuroanatomists, and the subsequent parcellations used, together with the multimodality data, to train a classifier for automatic delineation of similar borders on a validation set. The final group maximum probability map (MPM) parcellation was then formed from the individual probabilistic areal maps.

3 Results

We applied the proposed predictive framework independently on: three single subjects, three populations of five subjects, two populations of ten subjects, and one population of 20 subjects. For all analyses the same six subjects were held-out for prediction.

Figure 2, left panel, shows the impact of the amount of training data on the attainable upper and lower bounds. The figure shows the AUC curves for all the parcellations as more training subjects are included (different line styles). The results for single subjects show large uncertainty and predictive performance substantially below population based prediction (SBM: AUC=0.9486 (22), Ward: AUC=0.9615 (19), see Table 1: first column. Figures in parentheses give the standard deviation of the mean of the last digits across the different training networks), and hence training with a single subject is insufficient to characterize SC. However, already with five training subjects there is a large reduction in variability and increase in AUC (SBM: AUC=0.9790 (1), Ward: AUC=0.9799 (1)), and the inclusion of 10 (SBM: AUC=0.9831 (2), Ward: AUC=0.9833 (2)) and 20 subjects (SBM: AUC=0.9857, Ward: AUC=0.9857) only adds minor improvements (Table 1: columns 2 - 4). Furthermore, for five or more training subjects, the ranking of the atlases predictive performance remains constant, see Figure S7 in Online Resource 1. Consequently, even a limited sample of 20 training subjects provides sufficient robustness for predictive accuracy.

Using 20 subjects as our training population, the predictive assessment of the different parcellation schemes is investigated in Figure 2, right panel. The predictive performance of the three tested atlases are represented by the red symbols and the corresponding performance of the random (k-means) parcellation and data driven parcellations given by the blue, yellow and green curves respectively. With the number of parcels fixed to match those of the three tested atlases, we respectfully derived the following performances; for 68 parcels (Desikan-Killiany: AUC=0.9535, k-means: AUC=0.8886, Ward: AUC=0.9654, SBM: AUC=0.9701), for 148 parcels (Destrieux: AUC=0.9687, k-means: AUC=0.9153, Ward: AUC=0.9777, SBM: AUC=0.9787), and for 360 parcels

(HCP_MMP1.0: AUC=0.9807, k-means: AUC=0.9339, Ward: AUC=0.9836, SBM: AUC=0.9841), see Figure 2 (right panel). The vertical gap between the atlases and the random parcellations demonstrates that all three atlases perform far better than what would be expected by random if their parcels did not comply with the SC data. When considering the differences in predictive performance to the data-driven SC parcellations we find that both Desikan-Killiany (downward-pointing triangle) and Destrieux (upward-pointing triangle) have suboptimal performance. However, the HCP_MMP1.0 atlas (diamond) is not only superior to the surface morphology-based atlases but also almost on par with the best of the data-driven parcellations optimized to account for SC. We find these results to be robust to the applied threshold level and size of training population (considering at least five subjects for training), see Online Resource 1 (Figure S7).

To estimate the parcellation resolution supported by the SC data, we determined the beginning of the plateau regions (Figure 2, right panel: SBM, 700 parcels, green star; and Ward, 1000 parcels, yellow star), above which the predictive performance does not improve significantly (assessed using a paired t-test). These points can be interpreted as the minimum number of clusters required to sufficiently describe the SC data. The two data-driven models, SBM (green curves) and Ward clustering (yellow curves), show the same optimal predictive performance (AUC=0.9857). As they are based on different modeling approaches (SBM: Bayesian model with MCMC inference, Ward: deterministic agglomerative hierarchical clustering) this implies that the estimated upper bound is robust. In addition, both models approach the AUC of the best non-parametric link-predictor (shortest path, AUC=0.9875), shown as a black horizontal line, suggesting that no important information regarding the structural organization is lost when employing data-driven parcellations.

To investigate the effect of increasing the number of parcels to the maximum possible, we considered the limit where each node is given its own (singleton) cluster. This gave a much lower predictive performance (AUC=0.9336) than all considered atlases and SC parcellations.

Additional scores for single subjects and populations of 5 and 10 as well as scores obtained using standard non-parametric link-prediction methods [Liben-Nowell and Kleinberg, 2007] are given in Table 1 and in Online Resource 1 (Table S1).

3.1 Visualization of the parcellation structure

Figure 3 compares the parcellations from the three tested atlases together with the best performing SC parcellations using a population of 20 training subjects, both at the matching number of parcels and the SBM parcellation with 700 parcels, beyond which no significant improvement is found. Note how the parcellations found by SBM are spatially homogeneous even though the considered SBM does not incorporate any knowledge of spatial location. Additional parcellations are visualised in Online Resource 1, Figures S2-S4. The data-driven parcellations comply better with the existing atlases than the random parcellations (Online Resource 1, Figure S8).

4 Discussion

We here present a validation framework that permits quantitative assessment of any given parcellation scheme in the absence of a gold standard reference (ground truth parcellation). The framework uses statistical prediction to validate a parcellation by its ability to characterize the structure of independent brain connectivity data. Using this framework we validated three existing parcellations (not based on SC data) in their ability to characterize the organization of SC data.

Our framework, in being able to rank the performance of the prospective parcellations, shows that all three evaluated atlases are able to capture many of the features of SC and much better than would be expected by random. The framework further permits quantification of the improvement in predictive performance achieved by a recent multi-modality approach by Glasser et al. [Glasser et al., 2016] over those based solely upon surface morphology. In particular, we find that the multi-modality approach has performance almost on par with data-driven SC parcellations that are tailored to account for the organization of SC. These results are robust to the choice of streamline-count threshold applied to generate the SC networks and are consistent for the population based analyses, i.e. when at least five subjects are used in the training population.

As the three tested atlases perform far better than the lower-bound provided by the k-means random parcellations this implies that the organization of SC is in compliance with the atlases. The difference in predictive performance between the atlases and the estimated upper-bound for prediction given by the data-driven parcellations can be interpreted as the predictive loss due to the mismatch between SC and atlas parcel boundaries. As the two surface morphology-based atlases are unable to match the performance of the data-driven methods this implies reduced co-dependence between SC and surface morphology. However, the HCP_MMP1.0 atlas, being almost on par with the data-driven parcellations, emphasizes the utility of having multiple modalities.

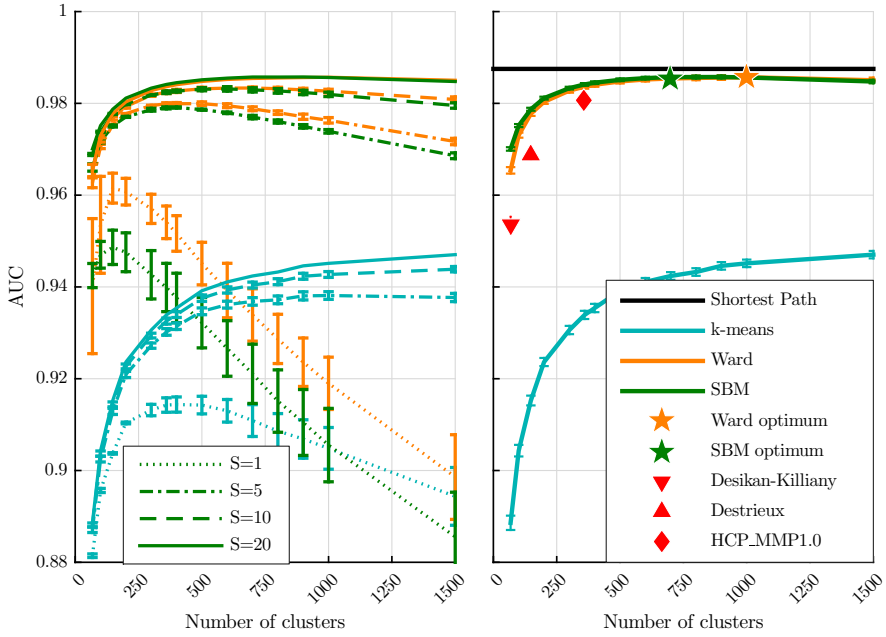


Fig. 2: Predictive performance measured by average AUC. Left panel: Predictive performance of the two data-driven SC parcellations as well as the random k-means parcellation (different colors), when using 1, 5, 10, and 20 subjects in the training set (different line styles). The error bars show the standard deviation of the mean across the training graphs. Right panel: Predictive performance using 20 training subjects of the three considered atlases: Desikan-Killiany (downward-pointing triangle), Destrieux (upward-pointing triangle), and HCP_MMP1.0 (diamond), as well as k-means random parcellations (blue) and data-driven SC parcellations Ward (yellow) and SBM (green). The stars indicate the optimal number of clusters, above which no significant increase in performance is observed for the two data-driven SC parcellations, based on a paired t-test. The predictive performance of shortest path is shown as a black horizontal line. The error bars show the standard deviation of the mean across the six test subjects.

4.1 Number of subjects

An important question answered by our framework is determining the number of subjects necessary to characterize SC data. Although it is recommended to use as many training subjects as possible, our validation framework demonstrates that even with limited data (20 training subjects) the predictive performance is sufficient to evaluate parcellations, and the ranking of the atlases remains constant for five or more subjects. Furthermore, the poor performance of data-driven SC parcellations when trained on a single subject emphasizes the importance of inference at the population level in order to sufficiently account for the organization of SC.

4.2 Parcellations preserve SC information

The best performing standard link prediction measure, shortest path, provides an estimate of the predictive performance that can be obtained taking all the SC information into account, as opposed to the aggregated information employed when prediction occurs at the level of parcels. The minor difference between the performance of shortest path and the data-driven parcellations implies that the latter are able to maintain the prominent information regarding the connectivity structure present in the data. We thereby find that the SC data is well represented using structural units defined by parcels, supporting analysis of SC at the level of structural units comprised of many vertices.

4.3 Number of parcels supported by SC data

In this work, we determined the optimum number of parcels supported by SC by locating the point beyond which no significant increase occurred in the data-driven SC parcellations. We found that the data-driven parcellations in general supported more parcels than the number specified in the recently proposed HCP_MMP1.0 atlas. Glasser et al. accordingly state that their parcellation may still underestimate the true number of parcels at the macroscale, as their subdivisioning of areas such as the primary visual cortex are coarser than reported previously [Glasser et al., 2016]. However, care must be taken when interpreting the estimated

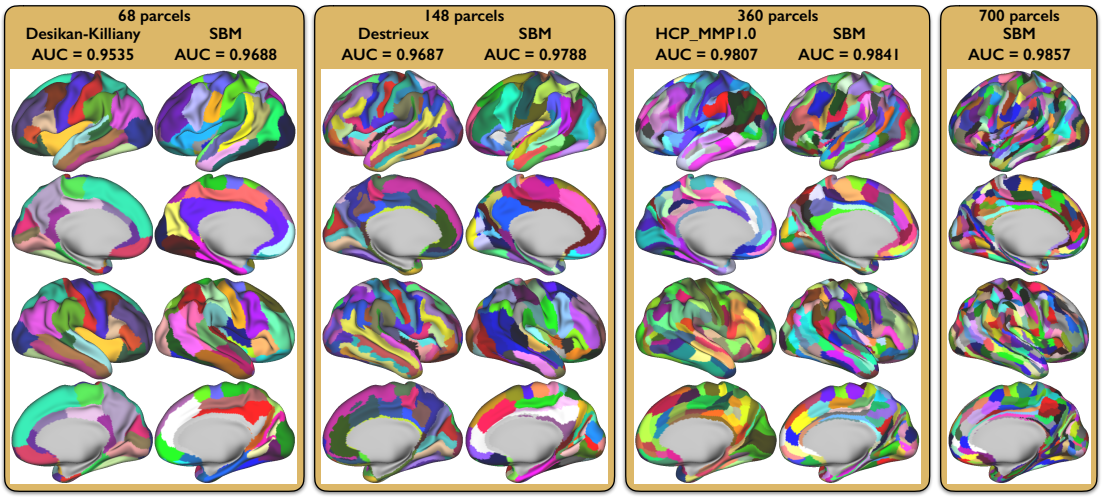


Fig. 3: Parcellations from the (left to right) Desikan-Killiany atlas and SBM with 68 parcels, the Destrieux atlas and SBM with 148 parcels, HCP_MMP1.0 and SBM with 360 parcels, and SBM with 700 parcels (green star in Figure 2 (right panel)), all visualised on the inflated surface. SBM was trained on 20 subjects. The AUC score is the average score across the six test subjects for the specific parcellation shown. The standard deviation of the mean is below 0.002 for all shown parcellations.

numbers of clusters supported by the SC data as we find that it is dependent upon the number of subjects included in the training data as well as the graph threshold level (see Online Resource S2 (Figure S7)). Furthermore, an exact estimate of the optimal number of clusters is non-trivial due to the broad range of resolutions that produce similar predictive performance. The optimum may also be influenced by biases in the SC data, as discussed below, that can potentially lead to overestimating the number of structural units. However, as the results for the k-means parcellations show, simply using a high number of clusters is not sufficient to capture the complexity of the SC data. Furthermore, we observed that the AUC did not continue to increase with more clusters. In particular, the extreme case where all the clustering models coincide (singleton clustering) exhibited poor predictive performance (AUC=0.9336 for 20 subjects). Even so, we observed that the performance of the singleton parcellations substantially improved when including more subjects in the training set, and we anticipate that with unlimited training data the averaging across training subjects may reduce the noise of the data to such an extent that the observed performance drop of singleton clusters may disappear. Thus, although our results point to the need for substantially more parcels than available in the considered atlases, these results may be heavily influenced by the level of noise and the biases, as discussed below.

4.4 Biases in the surface registration between subjects

A possible limitation to the results reported herein is the accuracy of the initial vertex-to-vertex registration framework, as provided by the HCP pipeline [Glasser et al., 2013]. As this is driven by surface topology [Fischl, 2012], there exists the possibility that the subsequent vertex alignment is biased towards anatomical landmarks (and therefore provides atlases based upon surface morphology with an inherent prediction boost). As anatomy may not be an optimal predictor of SC, this means that the assumed vertex-to-vertex correspondence may not fully reflect the nature of the SC data. Hence, such a bias would exhibit itself as noise in the vertex labeling, which would in turn propagate to the adjacency matrix (graph). As a consequence, it would be more difficult for a data-driven model to produce large homogeneous clusters of vertices which all possess similar patterns of SC. This would make larger clusters less likely, and so our predictive framework could therefore support an over-parcellation. Advanced vertex registration procedures, such as that employed in [Glasser et al., 2016], may improve matters as the imposed predictive bias will be balanced between multiple modalities.

4.5 Tractography biases

SC is established from dMRI data by integrating the derived local estimates of fibre bundle orientations obtained with standard tractography methods [Behrens et al., 2003a].

However, just as for all other methods that estimate connectivity, tractography has its own challenges and limitations, e.g. gyral crown bias [Van Essen et al., 2013a, Reveley et al., 2015, Donahue et al., 2016], which could affect the precise location of parcel borders, path length dependencies [Liptrot et al., 2014] and other factors which together are known to impose unknown levels of Type I and Type II errors on the estimated connections [Jones, 2010, Morris et al., 2008, Le Bihan et al., 2006, Jones et al., 2013]. These confounds, biases and shortcomings of tractography are as yet not fully quantifiable due to the lack of a gold-standard reference [Knösche et al., 2015], and indeed are not detectable as they will be present in both training and test datasets. Yet, despite all the challenges in tractography, we find that the existing atlases in general comply well with the SC data and that the best performing atlas is almost on par with the data-driven parcellations tailored for SC. This indeed points to compliance of the organization of SC with other modalities.

4.6 Other biases

No matter which connectivity modality is employed within our prediction framework, the inclusion of more training subjects, whilst increasing the signal-to-noise ratio, will not be able to compensate for modality-specific biases present in both training and test populations. However, the framework introduced herein can easily be extended to include multi-modality data such as fMRI, tracer studies, or histological reconstruction of axonal trajectories [Amunts and Zilles, 2015]. As demonstrated by Glasser et al. [Glasser et al., 2016], the incorporation of many independent data-sources can mollify the effects of their individual biases. Indeed, sufficiently many sources may even render the manually-intensive verification of parcels unnecessary.

4.7 Thresholding of SC networks

As with any graph model of connectivity, false positives and false negatives will occur as the incorrect presence or absence of links. Herein, as is common practice [Drakesmith et al., 2015, Hagmann et al., 2007, Hagmann et al., 2008], we attempt to remove many of the false positive connections by thresholding the SC graphs prior to modeling. However, this uniformly-applied strategy also increases the false negative rate. Unfortunately, whilst the false positive rate can be reduced to zero by increasing the threshold, the minimum false negative rate, achieved at null thresholding, will be non-zero and can only be improved by better data acquisition and processing strategies. As such, it must be noted that the chosen threshold level determines the balance between a model's specificity and sensitivity, and no optimal threshold exists [Knösche et al., 2015, Zalesky et al., 2016, Qi et al., 2015, Dyrby et al., 2007]. Even though the applied threshold of 200 streamline counts seems reasonable for this data set, as different initialisations of the tractography are able to predict each other very well (see Figure S1), the threshold is still arbitrarily chosen. However, as discussed earlier, the ranking of the considered atlases is maintained across all tested thresholds, demonstrating robustness of the proposed predictive validation procedure to the chosen threshold level.

4.8 Outlook

Our predictive validation procedure shows that the recently proposed HCP_MMP1.0 atlas provides a reasonable model of SC parcellation, and should be preferred to those based solely upon surface morphology. We validated parcellations using independent SC data, but the proposed validation framework is generic and therefore applicable to any other brain connectivity mapping approach. As the number of data-sources and data-derived approaches to structural and functional connectivity, and thereby also parcellation schemes, will only increase in the future, we foresee that the prediction framework presented herein will prove to be an important tool in assessing their quality.

Acknowledgements This project was supported by the Lundbeck Foundation, grant no. R105-9813. The Tesla K40 GPU card used for the BedpostX calculations was donated by the NVIDIA Corporation. Data were provided by the Human Connectome Project, WU-Minn Consortium (Principal Investigators: David Van Essen and Kamil Ugurbil; 1U54MH091657) funded by the 16 NIH Institutes and Centers that support the NIH Blueprint for Neuroscience Research; and by the McDonnell Center for Systems Neuroscience at Washington University.

References

[Ambrosen et al., 2014] Ambrosen, K. S., Albers, K. J., Dyrby, T. B., Schmidt, M. N., and Morup, M. (2014). Nonparametric bayesian clustering of structural whole brain connectivity in full image resolution. In *Pattern Recognition in Neuroimaging, 2014 International Workshop on*, pages 1–4. IEEE.

[Amunts and Zilles, 2015] Amunts, K. and Zilles, K. (2015). Architectonic mapping of the human brain beyond brodmann. *Neuron*, 88(6):1086–1107.

[Andersen et al., 2014] Andersen, K. W., Madsen, K. H., Siebner, H. R., Schmidt, M. N., Mørup, M., and Hansen, L. K. (2014). Non-parametric bayesian graph models reveal community structure in resting state fmri. *NeuroImage*, 100:301–315.

[Baldassano et al., 2015] Baldassano, C., Beck, D. M., and Fei-Fei, L. (2015). Parcellating connectivity in spatial maps. *PeerJ*, 3:e784.

[Behrens et al., 2007] Behrens, T., Berg, H. J., Jbabdi, S., Rushworth, M., and Woolrich, M. (2007). Probabilistic diffusion tractography with multiple fibre orientations: What can we gain? *Neuroimage*, 34(1):144–155.

[Behrens et al., 2003a] Behrens, T., Johansen-Berg, H., Woolrich, M., Smith, S., Wheeler-Kingshott, C., Boulby, P., Barker, G., Sillery, E., Sheehan, K., Ciccarelli, O., et al. (2003a). Non-invasive mapping of connections between human thalamus and cortex using diffusion imaging. *Nature neuroscience*, 6(7):750–757.

[Behrens et al., 2003b] Behrens, T., Woolrich, M., Jenkinson, M., Johansen-Berg, H., Nunes, R., Clare, S., Matthews, P., Brady, J., and Smith, S. (2003b). Characterization and propagation of uncertainty in diffusion-weighted mr imaging. *Magnetic resonance in medicine*, 50(5):1077–1088.

[Braitenberg and Schüz, 1991] Braitenberg, V. and Schüz, A. (1991). *Anatomy of the cortex: Statistics and geometry*. Springer-Verlag Publishing.

[Bullmore and Sporns, 2009] Bullmore, E. and Sporns, O. (2009). Complex brain networks: graph theoretical analysis of structural and functional systems. *Nature Reviews Neuroscience*, 10(3):186–198.

[Clauset et al., 2008] Clauset, A., Moore, C., and Newman, M. E. (2008). Hierarchical structure and the prediction of missing links in networks. *Nature*, 453(7191):98–101.

[Desikan et al., 2006] Desikan, R. S., Ségonne, F., Fischl, B., Quinn, B. T., Dickerson, B. C., Blacker, D., Buckner, R. L., Dale, A. M., Maguire, R. P., Hyman, B. T., et al. (2006). An automated labeling system for subdividing the human cerebral cortex on mri scans into gyral based regions of interest. *Neuroimage*, 31(3):968–980.

[Destrieux et al., 2010] Destrieux, C., Fischl, B., Dale, A., and Halgren, E. (2010). Automatic parcellation of human cortical gyri and sulci using standard anatomical nomenclature. *Neuroimage*, 53(1):1–15.

[Donahue et al., 2016] Donahue, C. J., Sotiropoulos, S. N., Jbabdi, S., Hernandez-Fernandez, M., Behrens, T. E., Dyrby, T. B., Coalson, T., Kennedy, H., Knoblauch, K., Van Essen, D. C., et al. (2016). Using diffusion tractography to predict cortical connection strength and distance: a quantitative comparison with tracers in the monkey. *The Journal of Neuroscience*, 36(25):6758–6770.

[Drakesmith et al., 2015] Drakesmith, M., Caeyenberghs, K., Dutt, A., Lewis, G., David, A., and Jones, D. (2015). Overcoming the effects of false positives and threshold bias in graph theoretical analyses of neuroimaging data. *NeuroImage*, 118:313–333.

[Dyrby et al., 2007] Dyrby, T. B., Søgaard, L. V., Parker, G. J., Alexander, D. C., Lind, N. M., Baaré, W. F., Hay-Schmidt, A., Eriksen, N., Pakkenberg, B., Paulson, O. B., et al. (2007). Validation of in vitro probabilistic tractography. *Neuroimage*, 37(4):1267–1277.

[Eickhoff et al., 2011] Eickhoff, S. B., Bzdok, D., Laird, A. R., Roski, C., Caspers, S., Zilles, K., and Fox, P. T. (2011). Co-activation patterns distinguish cortical modules, their connectivity and functional differentiation. *Neuroimage*, 57(3):938–949.

[Fan et al., 2016] Fan, L., Li, H., Zhuo, J., Zhang, Y., Wang, J., Chen, L., Yang, Z., Chu, C., Xie, S., Laird, A. R., et al. (2016). The human brainnetome atlas: A new brain atlas based on connectional architecture. *Cerebral Cortex*, page bhw157.

[Feinberg et al., 2010] Feinberg, D. A., Moeller, S., Smith, S. M., Auerbach, E., Ramanna, S., Glasser, M. F., Miller, K. L., Ugurbil, K., and Yacoub, E. (2010). Multiplexed echo planar imaging for sub-second whole brain fmri and fast diffusion imaging. *PloS one*, 5(12):e15710.

[Fischl, 2012] Fischl, B. (2012). Freesurfer. *Neuroimage*, 62(2):774–781.

[Fischl et al., 2004] Fischl, B., van der Kouwe, A., Destrieux, C., Halgren, E., Ségonne, F., Salat, D. H., Busa, E., Seidman, L. J., Goldstein, J., Kennedy, D., et al. (2004). Automatically parcellating the human cerebral cortex. *Cerebral cortex*, 14(1):11–22.

[Fornito et al., 2010] Fornito, A., Zalesky, A., and Bullmore, E. T. (2010). Network scaling effects in graph analytic studies of human resting-state fmri data. *Frontiers in systems neuroscience*, 4.

[Glasser et al., 2016] Glasser, M. F., Coalson, T. S., Robinson, E. C., Hacker, C. D., Harwell, J., Yacoub, E., Ugurbil, K., Andersson, J., Beckmann, C. F., Jenkinson, M., Smith, S. M., and Van Essen, D. C. (2016). A multi-modal parcellation of human cerebral cortex. *Nature*, 536(7615):171–178.

[Glasser et al., 2013] Glasser, M. F., Sotiropoulos, S. N., Wilson, J. A., Coalson, T. S., Fischl, B., Andersson, J. L., Xu, J., Jbabdi, S., Webster, M., Polimeni, J. R., et al. (2013). The minimal preprocessing pipelines for the human connectome project. *Neuroimage*, 80:105–124.

[Gong et al., 2009] Gong, G., He, Y., Concha, L., Lebel, C., Gross, D. W., Evans, A. C., and Beaulieu, C. (2009). Mapping anatomical connectivity patterns of human cerebral cortex using in vivo diffusion tensor imaging tractography. *Cerebral cortex*, 19(3):524–536.

[Hagmann, 2005] Hagmann, P. (2005). From diffusion mri to brain connectomics [phd thesis]. *Lausanne: Ecole Polytechnique Fdrale de Lausanne (EPFL).*, 127 p.

[Hagmann et al., 2008] Hagmann, P., Cammoun, L., Gigandet, X., Meuli, R., Honey, C. J., Wedeen, V. J., and Sporns, O. (2008). Mapping the structural core of human cerebral cortex. *PLoS biology*, 6(7):e159.

[Hagmann et al., 2007] Hagmann, P., Kurant, M., Gigandet, X., Thiran, P., Wedeen, V. J., Meuli, R., and Thiran, J.-P. (2007). Mapping human whole-brain structural networks with diffusion mri. *PloS one*, 2(7):e597.

[Hernández et al., 2013] Hernández, M., Guerrero, G. D., Cecilia, J. M., García, J. M., Inuggi, A., Jbabdi, S., Behrens, T. E., and Sotiropoulos, S. N. (2013). Accelerating fibre orientation estimation from diffusion weighted magnetic resonance imaging using gpus. *PLoS One*, 8(4):e61892.

[Holland et al., 1983] Holland, P. W., Laskey, K. B., and Leinhardt, S. (1983). Stochastic blockmodels: First steps. *Social networks*, 5(2):109–137.

[Jenkinson et al., 2012] Jenkinson, M., Beckmann, C. F., Behrens, T. E., Woolrich, M. W., and Smith, S. M. (2012). Fsl. *Neuroimage*, 62(2):782–790.

[Jones, 2010] Jones, D. K. (2010). Challenges and limitations of quantifying brain connectivity in vivo with diffusion mri. *Imaging in Medicine*, 2(3):341–355.

[Jones et al., 2013] Jones, D. K., Knösche, T. R., and Turner, R. (2013). White matter integrity, fiber count, and other fallacies: the do's and don'ts of diffusion mri. *Neuroimage*, 73:239–254.

[Knösche et al., 2015] Knösche, T. R., Anwander, A., Liptrot, M., and Dyrby, T. B. (2015). Validation of tractography: comparison with manganese tracing. *Human brain mapping*, 36(10):4116–4134.

[Le Bihan et al., 2006] Le Bihan, D., Poupon, C., Amadon, A., and Lethimonnier, F. (2006). Artifacts and pitfalls in diffusion mri. *Journal of magnetic resonance imaging*, 24(3):478–488.

[Liben-Nowell and Kleinberg, 2007] Liben-Nowell, D. and Kleinberg, J. (2007). The link-prediction problem for social networks. *Journal of the American society for information science and technology*, 58(7):1019–1031.

[Liptrot et al., 2014] Liptrot, M. G., Sidaros, K., and Dyrby, T. B. (2014). Addressing the path-length-dependency confound in white matter tract segmentation. *PloS one*, 9(5):e96247.

[MacQueen et al., 1967] MacQueen, J. et al. (1967). Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, volume 1, pages 281–297. Oakland, CA, USA.

[Milchenko and Marcus, 2013] Milchenko, M. and Marcus, D. (2013). Obscuring surface anatomy in volumetric imaging data. *Neuroinformatics*, 11(1):65–75.

[Miller et al., 2009] Miller, K., Jordan, M. I., and Griffiths, T. L. (2009). Nonparametric latent feature models for link prediction. In *Advances in neural information processing systems*, pages 1276–1284.

[Moeller et al., 2010] Moeller, S., Yacoub, E., Olman, C. A., Auerbach, E., Strupp, J., Harel, N., and Uğurbil, K. (2010). Multiband multislice ge-epi at 7 tesla, with 16-fold acceleration using partial parallel imaging with application to high spatial and temporal whole-brain fmri. *Magnetic Resonance in Medicine*, 63(5):1144–1153.

[Morris et al., 2008] Morris, D. M., Embleton, K. V., and Parker, G. J. (2008). Probabilistic fibre tracking: differentiation of connections from chance events. *Neuroimage*, 42(4):1329–1339.

[Murre and Sturdy, 1995] Murre, J. and Sturdy, D. (1995). The connectivity of the brain: multi-level quantitative analysis. *Biological cybernetics*, 73(6):529–545.

[Nowicki and Snijders, 2001] Nowicki, K. and Snijders, T. A. B. (2001). Estimation and prediction for stochastic blockstructures. *Journal of the American Statistical Association*, 96(455):1077–1087.

[Parisot et al., 2016] Parisot, S., Arslan, S., Passerat-Palmbach, J., Wells, W. M., and Rueckert, D. (2016). Group-wise parcellation of the cortex through multi-scale spectral clustering. *NeuroImage*.

[Passingham et al., 2002] Passingham, R. E., Stephan, K. E., and Kötter, R. (2002). The anatomical basis of functional localization in the cortex. *Nature Reviews Neuroscience*, 3(8):606–616.

[Qi et al., 2015] Qi, S., Meesters, S., Nicolay, K., ter Haar Romeny, B. M., and Ossenblok, P. (2015). The influence of construction methodology on structural brain network measures: a review. *Journal of neuroscience methods*, 253:170–182.

[Reveley et al., 2015] Reveley, C., Seth, A. K., Pierpaoli, C., Silva, A. C., Yu, D., Saunders, R. C., Leopold, D. A., and Frank, Q. Y. (2015). Superficial white matter fiber systems impede detection of long-range cortical connections in diffusion mr tractography. *Proceedings of the National Academy of Sciences*, 112(21):E2820–E2828.

[Schmidt and Mørup, 2013] Schmidt, M. N. and Mørup, M. (2013). Nonparametric bayesian modeling of complex networks: An introduction. *Signal Processing Magazine, IEEE*, 30(3):110–128.

[Setsompop et al., 2012] Setsompop, K., Gagoski, B. A., Polimeni, J. R., Witzel, T., Wedeen, V. J., and Wald, L. L. (2012). Blipped-controlled aliasing in parallel imaging for simultaneous multislice echo planar imaging with reduced g-factor penalty. *Magnetic Resonance in Medicine*, 67(5):1210–1224.

[Smith, 2013] Smith, S. (2013). Introduction to the neuroimage special issue mapping the connectome. *NeuroImage*, 80(Complete).

[Sotiropoulos et al., 2013] Sotiropoulos, S., Moeller, S., Jbabdi, S., Xu, J., Andersson, J., Auerbach, E., Yacoub, E., Feinberg, D., Setsompop, K., Wald, L., et al. (2013). Effects of image reconstruction on fiber orientation mapping from multichannel diffusion mri: reducing the noise floor using sense. *Magnetic resonance in medicine*, 70(6):1682–1689.

[Sporns, 2012] Sporns, O. (2012). *Discovering the human connectome*. MIT press.

[Sporns et al., 2005] Sporns, O., Tononi, G., and Kötter, R. (2005). The human connectome: a structural description of the human brain. *PLoS computational biology*, 1(4):e42.

[Thirion et al., 2014] Thirion, B., Varoquaux, G., Dohmatob, E., and Poline, J.-B. (2014). Which fmri clustering gives good brain parcellations? *Frontiers in neuroscience*, 8(167):13.

[Tzourio-Mazoyer et al., 2002] Tzourio-Mazoyer, N., Landeau, B., Papathanassiou, D., Crivello, F., Etard, O., Delcroix, N., Mazoyer, B., and Joliot, M. (2002). Automated anatomical labeling of activations in spm using a macroscopic anatomical parcellation of the mni mri single-subject brain. *Neuroimage*, 15(1):273–289.

[Van Essen et al., 2013a] Van Essen, D. C., Jbabdi, S., Sotiropoulos, S. N., Chen, C., Dikranian, K., Coalson, T., Harwell, J., Behrens, T. E., and Glasser, M. F. (2013a). Mapping connections in humans and nonhuman primates: aspirations and challenges for diffusion imaging. *Diffusion MRI, 2nd edition (eds. Johansen-Berg, H. & Behrens, TEJ)*, pages 337–358.

[Van Essen et al., 2013b] Van Essen, D. C., Smith, S. M., Barch, D. M., Behrens, T. E., Yacoub, E., and Ugurbil, K. (2013b). The wu-minn human connectome project: an overview. *Neuroimage*, 80:62–79.

[Van Essen et al., 2012] Van Essen, D. C., Ugurbil, K., Auerbach, E., Barch, D., Behrens, T., Bucholz, R., Chang, A., Chen, L., Corbetta, M., Curtiss, S. W., et al. (2012). The human connectome project: a data acquisition perspective. *Neuroimage*, 62(4):2222–2231.

[Ward Jr, 1963] Ward Jr, J. H. (1963). Hierarchical grouping to optimize an objective function. *Journal of the American statistical association*, 58(301):236–244.

[White et al., 1976] White, H. C., Boorman, S. A., and Breiger, R. L. (1976). Social structure from multiple networks. i. blockmodels of roles and positions. *American journal of sociology*, pages 730–780.

[Xu et al., 2012] Xu, J., Moeller, S., Strupp, J., Auerbach, E., Chen, L., Feinberg, D., Ugurbil, K., and Yacoub, E. (2012). Highly accelerated whole brain imaging using aligned-blipped-controlled-aliasing multiband epi. In *Proceedings of the 20th Annual Meeting of ISMRM*, volume 2306.

[Zalesky et al., 2016] Zalesky, A., Fornito, A., Cocchi, L., Gollo, L. L., van den Heuvel, M. P., and Breakspear, M. (2016). Connectome sensitivity or specificity: which is more important? *NeuroImage*.

[Zalesky et al., 2010] Zalesky, A., Fornito, A., Harding, I. H., Cocchi, L., Yücel, M., Pantelis, C., and Bullmore, E. T. (2010). Whole-brain anatomical networks: does the choice of nodes matter? *Neuroimage*, 50(3):970–983.

Table 1: Average AUC across six test subjects and five restarts when predicting unseen connectivity graphs using data-driven parcellations including the singleton parcellation in which each node is given its own cluster, as well as the best performing non-parametric link predictor (shortest path) and the three considered atlases. The scores are given for the optimal number of parcels (Online Resource 1, Figure S7). For k-means clustering, the single subject score is given for 360 parcels, while remaining scores are given for 1000 parcels. The standard deviation of the mean on the last digits, across different training networks, is given in parentheses.

	Population size			
	Single (n=3)	5 (n=3)	10 (n=2)	20 (n=1)
UPPER BOUND ESTIMATES				
Shortest path	0.9453 (25)	0.9830 (3)	0.9862 (2)	0.9875
SBM	0.9486 (22)	0.9790 (1)	0.9831 (2)	0.9857
Ward clustering	0.9615 (19)	0.9799 (1)	0.9833 (2)	0.9857
BRAIN ATLASES				
HCP_MMP1.0	0.9595 (9)	0.9777 (1)	0.9796 (5)	0.9807
Destrieux	0.9599 (9)	0.9670 (2)	0.9681 (9)	0.9687
Desikan-Killiany	0.9479 (7)	0.9524 (6)	0.9530 (14)	0.9535
LOWER BOUND ESTIMATES				
k-means	0.9143 (9)	0.9381 (5)	0.9427 (4)	0.9451
Singleton	0.7027 (120)	0.8541 (27)	0.9016 (1)	0.9336

S1 Supplementary material

S1.1 Threshold

The structural connectivity graphs are binarised by zeroing everything below a chosen threshold. If the threshold chosen is too low the connectivity graphs are dominated by false positives. On the other hand, if a too high threshold is chosen then true connections are removed, leading to many false negatives. As probabilistic tractography is a probabilistic process, re-running the tractography on the same dataset gives a slightly different result. To investigate the effect of the chosen threshold and to find the optimum, tractography was re-run on two subjects and the connectivity graphs were created. The AUC between re-runs of the tractography were calculated for a range of thresholds between zero and 5300 counts. Figure S1 shows how well the runs pairwise predict each other for different thresholds. For low thresholds the performance is low due to all the false positives in the graphs. Around a threshold of 200 the predictive performance stabilises with only a small increase in the performance for higher thresholds. Based on this result we chose a threshold of 200. To investigate the effect of our choice we also ran all analyses with a threshold of 50 and 1000. This method to find the threshold of the graphs can be applied in other studies, but the specific threshold will depend upon the number of streamlines seeded per voxel, the resolution of the data and parameters of the tractography method.

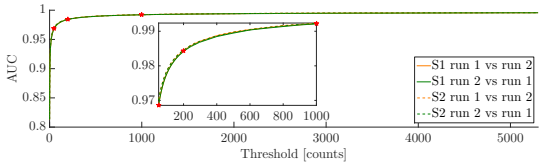


Fig. S1: AUC vs. threshold. For two subjects (S1 and S2) the tractography procudure was performed twice with different initialisation. The figure shows how well the two runs predict each other for a range of thresholds between zero and 5300 measured by AUC. The red stars indicate the tested thresholds.

S1.2 Inference in the stochastic block model

To infer the clusters in the stochastic block model (SBM) we seek a partition $\boldsymbol{z}$ of $\boldsymbol{A}^1,\ldots,\boldsymbol{A}^s$ into K clusters of nodes with similar structural connectivity pattern. Let $\boldsymbol{\pi}$ denote the probability distribution of any node belonging to the individual clusters, such that $p(z_i=k|\boldsymbol{\pi})=\pi_k$, where $\boldsymbol{\pi}=\{\pi_1,...,\pi_K\}$ and $\sum_{k=1}^K \pi_k = 1$.

To allow flexible cluster sizes, $\boldsymbol{\pi}$ is considered generated from a Dirichlet distribution:

$$p(\boldsymbol{\pi}|\boldsymbol{\alpha}) = \frac{1}{\mathrm{B}(\boldsymbol{\alpha})}\prod_{k=1}^{K}\pi_k^{\alpha_k-1}, \tag{S10}$$

where B is the multivariate beta function:

$$\mathrm{B}(\boldsymbol{\alpha}) = \frac{\prod_{k=1}^K \Gamma(\alpha_k)}{\Gamma(\sum_{k=1}^K \alpha_k)}. \tag{S11}$$

This reveals the following joint prior over $\boldsymbol{z}$ and $\boldsymbol{\pi}$:

$$p(\boldsymbol{\pi},\boldsymbol{z}|\boldsymbol{\alpha}) \;=\; p(\boldsymbol{\pi}|\boldsymbol{\alpha})\prod_{i=1}^{J}p(z_i|\boldsymbol{\pi}) \;=\; \frac{1}{\mathrm{B}(\boldsymbol{\alpha})}\prod_{k=1}^{K}\pi_k^{m_k+\alpha_k-1}, \tag{S12}$$

where m_k denotes the number of nodes belonging to cluster k, such that $\sum_{k=1}^K m_k = J$.

Imposing equal concentration parameter on all clusters $\frac{C}{K}=\alpha_1=\ldots=\alpha_K$ and marginalizing over $\boldsymbol{\pi}$ we obtain the effective prior over $\boldsymbol{z}$, resulting in a so-called multivariate Pólya distribution:

$$p(\boldsymbol{z}|C) \;=\; \int p(\boldsymbol{\pi},\boldsymbol{z}|C)d\boldsymbol{\pi} \;=\; \frac{\Gamma(C)}{\Gamma(C+J)}\prod_{k=1}^{K}\frac{\Gamma(\frac{C}{K}+m_k)}{\Gamma(\frac{C}{K})} \tag{S13}$$

For a given partition $\boldsymbol{z}$ the prior distribution on the probability η_{lm} of observing a link between nodes of cluster l and cluster m is imposed using the Beta distribution:

$$p(\eta_{lm}|\beta^+,\beta^-) = \frac{\Gamma(\beta^+ + \beta^-)}{\Gamma(\beta^+)\Gamma(\beta^-)}\eta_{lm}^{\beta^+-1}(1-\eta_{lm})^{\beta^--1}.$$

The probability of observing a link between node i and j for subject s follows the Bernoulli distribution such that the likelihood of $\mathscr{A} = \{\boldsymbol{A}^1,\ldots,\boldsymbol{A}^S\}$ is given by (see also [Andersen et al., 2014]):

$$p(\mathscr{A}|\boldsymbol{\eta},\boldsymbol{z}) = \prod_{s=1}^{S}\prod_{i>j}\eta_{z_iz_j}^{A^s_{ij}}(1-\eta_{z_iz_j})^{1-A^s_{ij}} = \prod_{l\geq m}\eta_{lm}^{N^+_{lm}}(1-\eta_{lm})^{N^-_{lm}}, \tag{S14}$$

where N^+_{lm} and N^-_{lm} respectively denotes the sum of all links and non-links between cluster l and m for all graphs in the population.

The conjugacy of the Beta prior and Bernoulli likelihood allows $\boldsymbol{\eta}$ to be analytically marginalized, revealing the following joint distribution:

$$\begin{aligned} p(\boldsymbol{A},\boldsymbol{z}|C,\beta^+,\beta^-) \;&=\; \int p(\boldsymbol{z}|C)\cdot p(\mathscr{A}|\boldsymbol{\eta},\boldsymbol{z})\cdot\prod_{l\geq m}p(\eta_{lm}|\beta^+,\beta^-)\,d\boldsymbol{\eta} \\ &=\; p(\boldsymbol{z}|C)\cdot\prod_{l\geq m}\frac{B(N^+_{lm}+\beta^+,N^-_{lm}+\beta^-)}{B(\beta^+,\beta^-)}, \end{aligned} \tag{S15}$$

where B denotes the beta function:

$$B(a,b) = \int \theta^{a-1}(1-\theta)^{b-1}d\theta = \frac{\Gamma(a)\Gamma(b)}{\Gamma(a+b)}$$

For a particular data set the model parameters are inferred using a sequence of independent Markov Chain Monte Carlo methods to sample from the posterior distribution.

The clustering is inferred using a combination of full and restricted Gibbs sampling procedures. In the full Gibbs sampling procedure, each node i is in turn proposed to be re-assigned, based on the posterior distribution of the single node assignment, obtained using Bayes' theorem for equation S15:

$$p(z_i=l|\mathscr{A},\boldsymbol{z}^{\setminus i},\beta^+,\beta^-,C) = \frac{p(\mathscr{A},\boldsymbol{z}^{\setminus i},z_i=l|\beta^+,\beta^-,C)}{\sum_m^K p(\mathscr{A},\boldsymbol{z}^{\setminus i},z_i=m|\beta^+,\beta^-,C)} \tag{S16}$$

where $\boldsymbol{z}^{\setminus i}$ denotes the cluster assignments for all nodes ignoring node i

In the restricted Gibbs sampling procedure, two clusters are randomly selected and three Gibbs sweeps are conducted, restricted to re-partitioning the nodes within the two selected clusters.

The three hyperparameters β^+,β^-,C are sampled individually using a Metropolis-Hastings procedure, where proposals are drawn from a Gaussian distribution with variance 1, centered at the current value of the parameter.

For all results in the paper, the following sampling procedure was utilized: one complete Gibbs sweep over all nodes followed by K restricted Gibbs proposals, followed by 10 Metropolis-Hastings proposals for each of the hyperparameters. A total of 100 sweeps of the above sampling procedure was performed. Following the last sweep of the MCMC sampling, the clustering was optimized towards a local posterior maximum using a hill-climbing procedure to repeatedly reassign the nodes one at a time to the cluster resulting in the highest posterior gain.

S1.3 Reliability estimation by Mutual Information

In order to quantify similarity between two partitions $\boldsymbol{z}$ and $\boldsymbol{z'}$ we use normalized mutual information (NMI), defined as:

$$NMI(\boldsymbol{z},\boldsymbol{z'}) = \frac{2 \cdot MI(\boldsymbol{z},\boldsymbol{z'})}{MI(\boldsymbol{z},\boldsymbol{z}) + MI(\boldsymbol{z'},\boldsymbol{z'})},$$

where the mutual information (MI) is given as:

$$MI(\boldsymbol{z},\boldsymbol{z'}) = \sum_{kk'} P(k,k') \log\left(\frac{P(k,k')}{P(k)P(k')}\right),$$

with $P(k,k')$ being the probability that a node in cluster k in the first partition is in cluster k' in the second partition. NMI takes values between zero and one where one indicates that a permutation of the groups exists such that the partitions are identical, and zero indicates that the partitions are perfectly independent.

S2 Supplementary figures

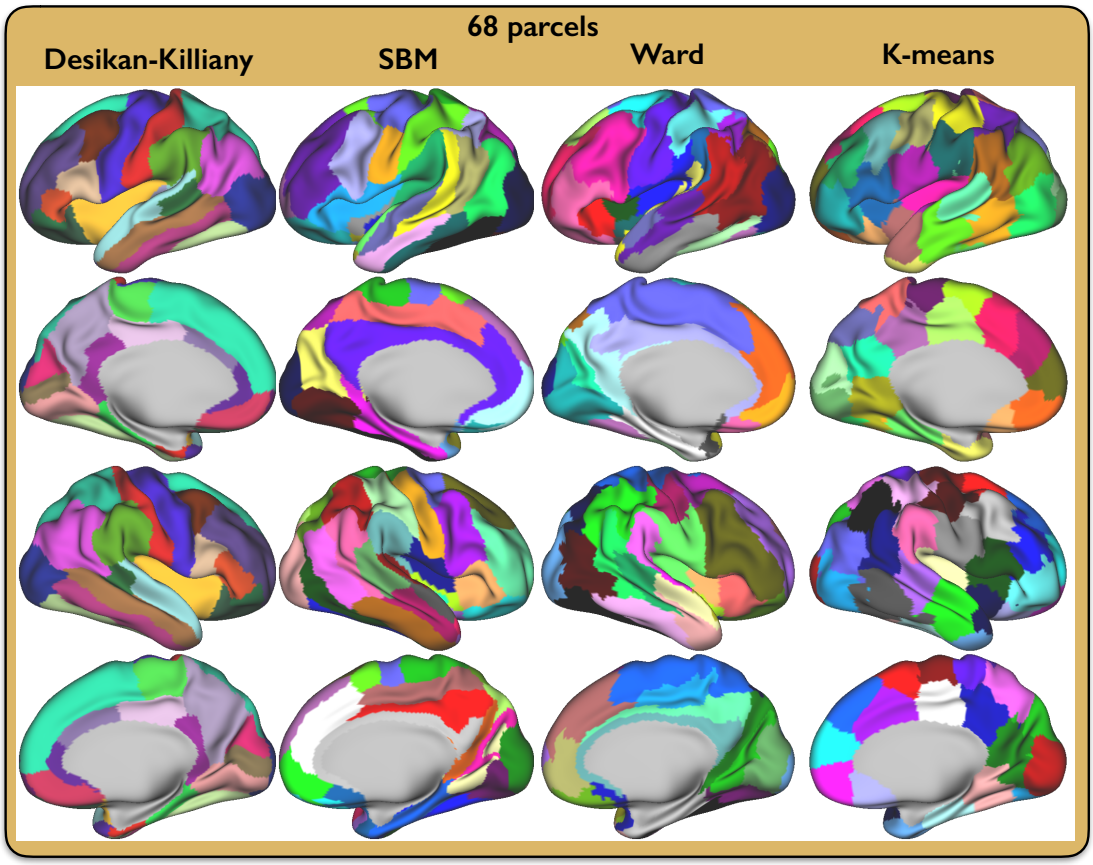


Fig. S2: Parcellations with 68 clusters for a population of 20 subjects and a threshold of 200, shown on the inflated surface. From left: The Desikan-Killiany atlas, SBM parcellation, Ward clustering, and K-means clustering.

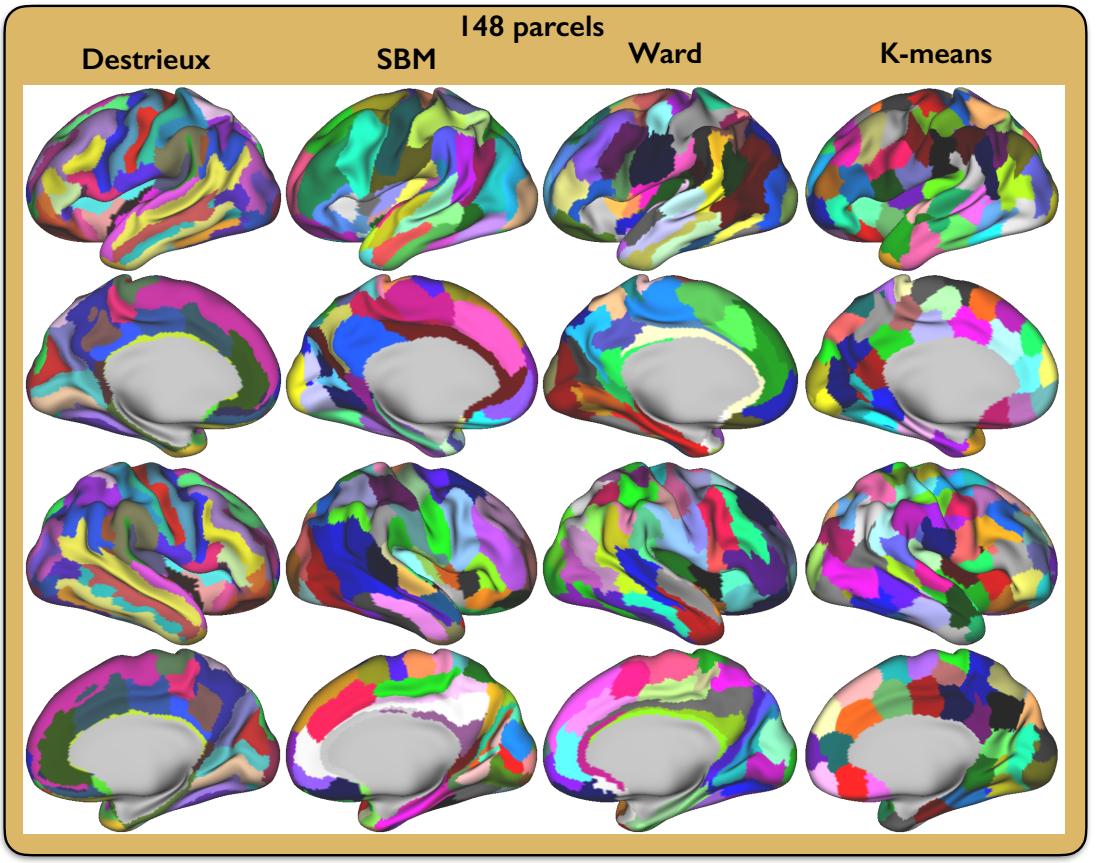


Fig. S3: Parcellations with 148 clusters for a population of 20 subjects and a threshold of 200, shown on the inflated surface. From left: The Destrieux atlas, SBM parcellation, Ward clustering, and K-means clustering.

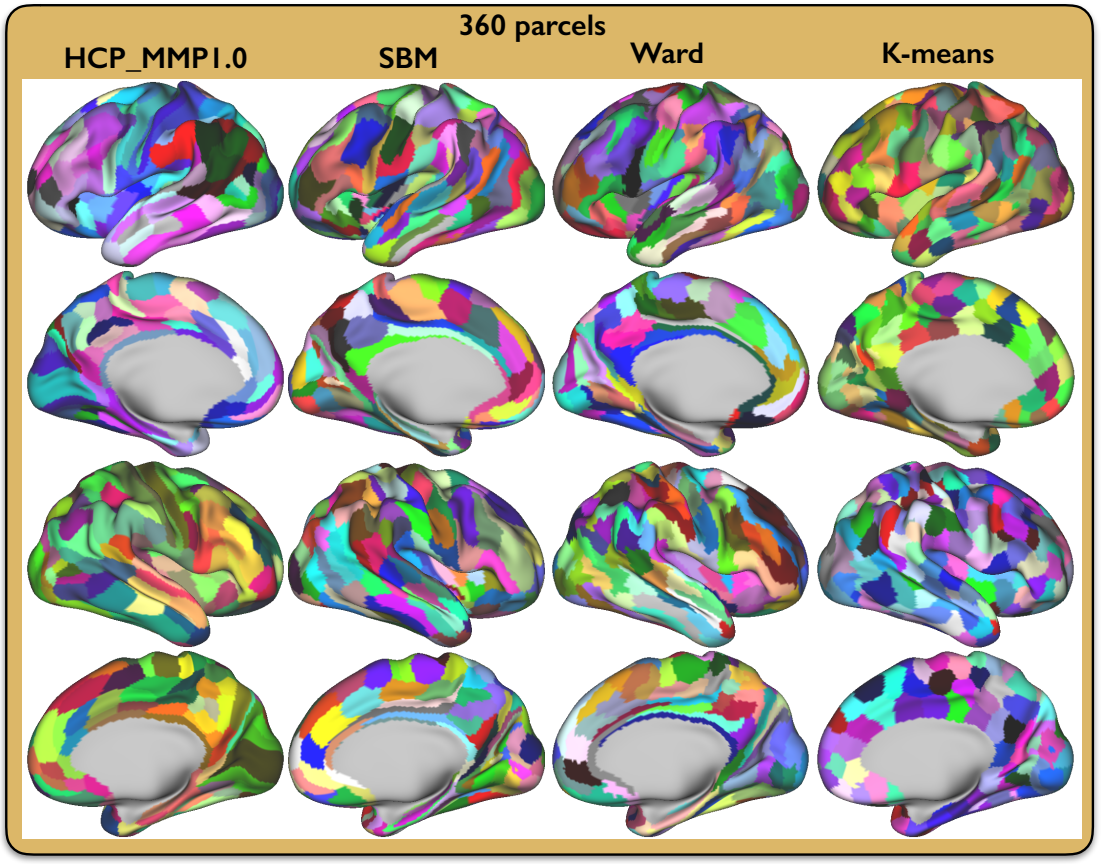
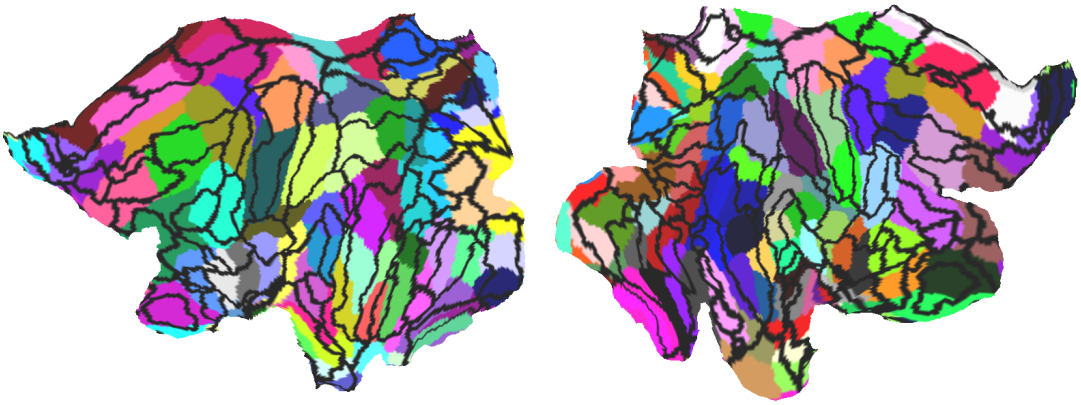
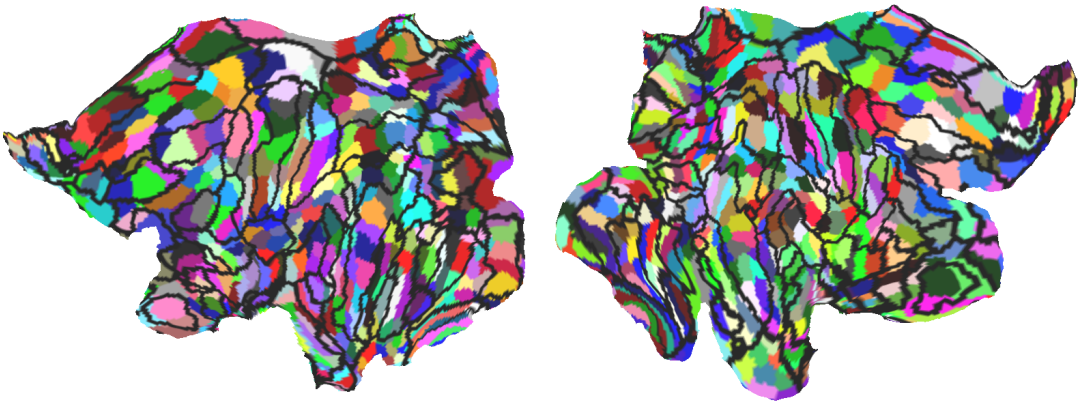


Fig. S4: Parcellations with 360 clusters for a population of 20 subjects and a threshold of 200, shown on the inflated surface. From left: The Human Connectome Project multi-modal parcellation (HCP_MMP1.0), SBM parcellation, Ward clustering, and K-means clustering.

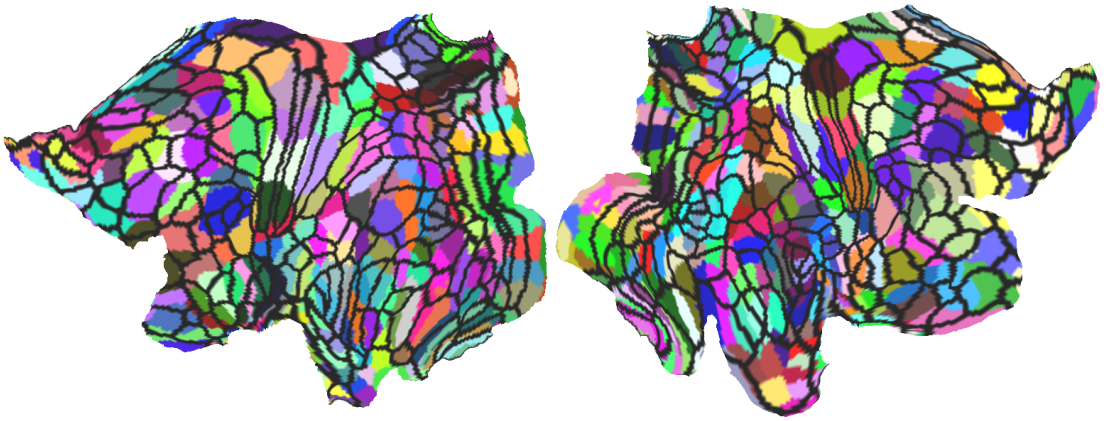


(a)

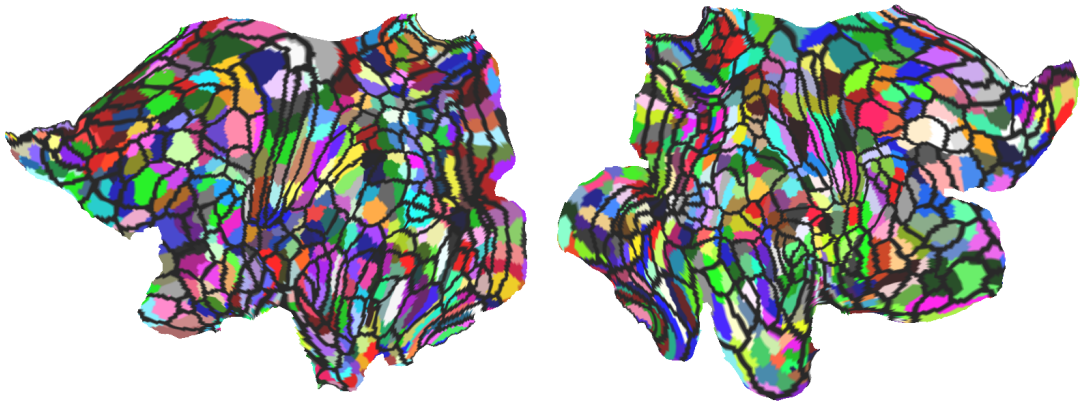


(b)

Fig. S5: Flatmaps of the SBM parcellations with (a) 148 clusters and (b) 700 clusters overlaid with the borders of the Destrieux atlas.



(a)



(b)

Fig. S6: Flatmaps of the SBM parcellations with (a) 360 clusters and (b) 700 clusters overlaid with the borders of the HCP_MMP1.0 atlas.

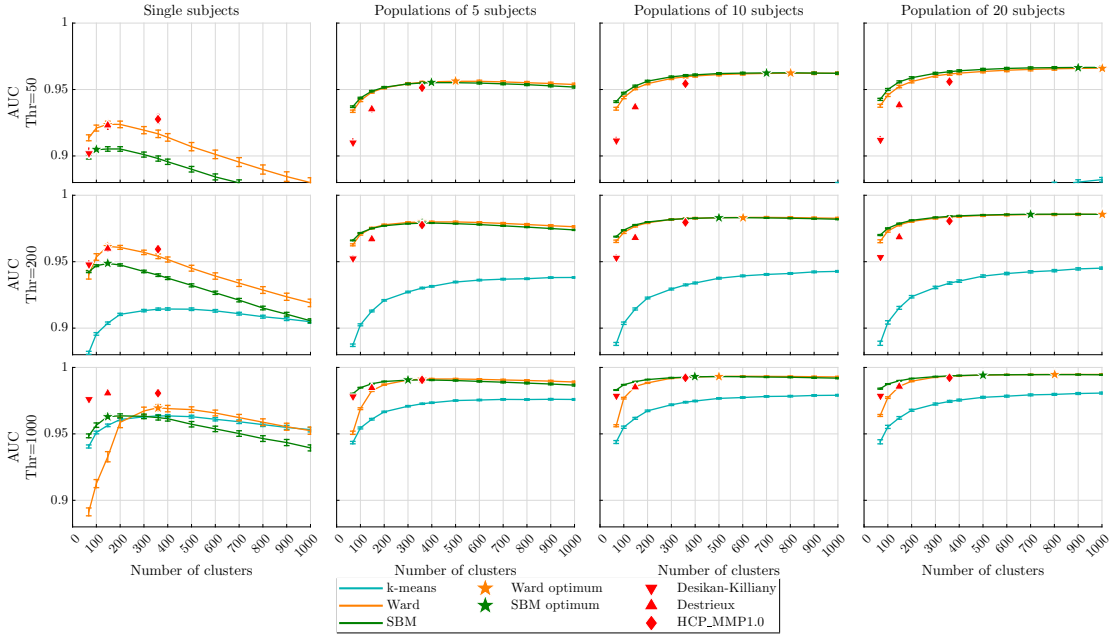


Fig. S7: Average AUC across the six test subjects vs the number of clusters for thresholds of 50, 200 and 1000, and population sizes of 1, 5, 10 and 20. The error bars indicate the standard deviation of the mean across the six test subjects. The stars mark the optimal number of clusters for the stochastic block model (green) and ward clustering (yellow), as no significant increase in performance is observed by using more clusters. The optimal number of clusters are found using a paired t-test.

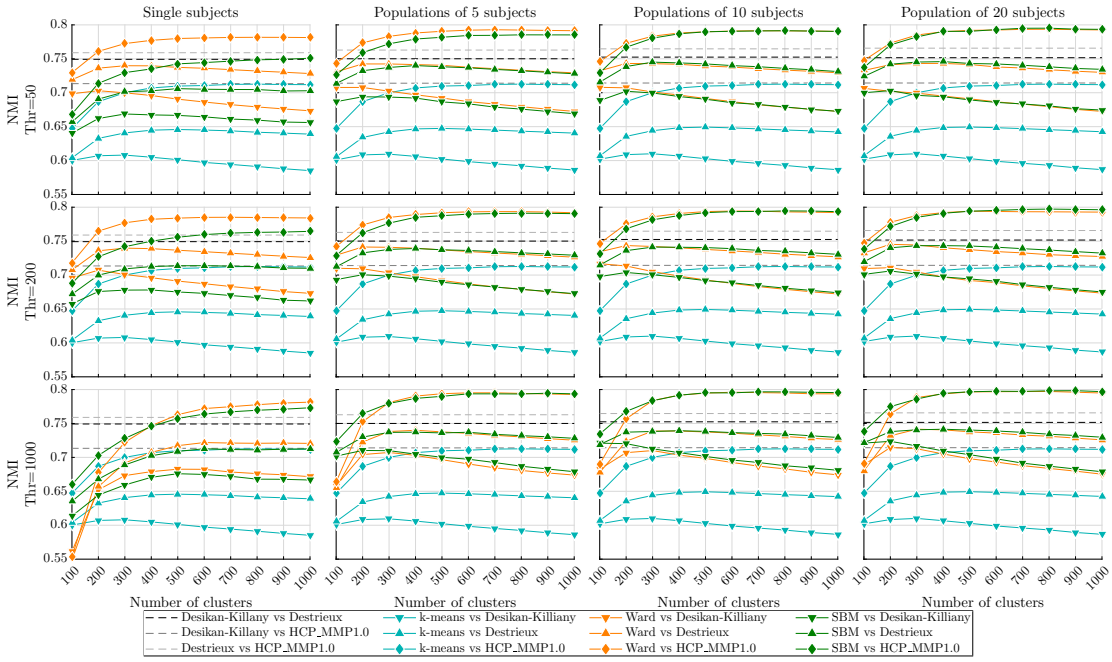


Fig. S8: Normalised mutual information (NMI) vs the number of clusters for thresholds of 50, 200 and 1000, and populations of of 1, 5, 10 and 20 subjects. Dashed lines show the reliability of the methods, NMI between Desikan-Killiany and Destrieux (black), between Desikan-Killiany and HCP_MMP1.0 (gray), and between Destrieux and HCP_MMP1.0 (light gray). Solid lines show NMI between the three parcellation approaches, the stochastic block model (green), ward clustering (yellow) and k-means clustering (blue), and Desikan-Killiany (downward-pointing triangles), Destrieux (upward-pointing triangles) and HCP_MMP1.0 (diamonds).

S3 Supplementary tables

Table S1: Average AUC across six test subjects and five restarts when predicting unseen connectivity graphs using data-driven parcellations including the singleton parcellation in which each node is given its own cluster, as well as the considered non-parametric link predictors and the three considered atlases. The scores are given for the optimal number of parcels (SI, Figure S7). For k-means clustering, the single subject score is given for 360 parcels, while remaining scores are given for 1000 parcels. In parentheses is given the standard deviation of the mean on last digit across different training networks.

	Population size			
	Single (n=3)	5 (n=3)	10 (n=2)	20 (n=1)
SBM	0.9486 (22)	0.9790 (1)	0.9831 (2)	0.9857
Ward clustering	0.9615 (19)	0.9799 (1)	0.9833 (2)	0.9857
k-means	0.9143 (9)	0.9381 (5)	0.9427 (4)	0.9451
Singleton	0.7027 (120)	0.8541 (27)	0.9016 (1)	0.9336
Shortest path	0.9453 (25)	0.9830 (3)	0.9862 (2)	0.9875
Common neighbor	0.9339 (63)	0.9792 (9)	0.9843 (4)	0.9865
Jaccard	0.9368 (64)	0.9816 (2)	0.9855 (2)	0.9874
Adamic/Adar	0.9346 (63)	0.9798 (8)	0.9848 (3)	0.9869
Preferential attach.	0.5756 (25)	0.6100 (6)	0.6188 (2)	0.6247
HCP_MMP1.0	0.9595 (9)	0.9777 (1)	0.9796 (5)	0.9807
Destrieux	0.9599 (9)	0.9670 (2)	0.9681 (9)	0.9687
Desikan-Killiany	0.9479 (7)	0.9524 (6)	0.9530 (14)	0.9535

Appendix F

Functional Whole-Brain Parcellation Improved by the Inclusion of Structural Connectivity

---

Functional Whole-Brain Parcellation Improved by the Inclusion of Structural Connectivity. Kristoffer Jon Albers, Karen Sandø Ambrosen, Rasmus Røge, Matthew G. Liptrot, Kasper Winther Andersen, Hartwig R. Siebner, Tim B. Dyrby, Kristoffer H. Madsen, Mikkel N. Schmidt, and Morten Mørup. (preliminary work).

Functional Whole-Brain Parcellation Improved by the Inclusion of Structural Connectivity

Kristoffer Jon Albers*, Karen Sandø Ambrosen*†, Rasmus Røge*, Matthew G. Liptrot*,
Tue Herlau*, Kasper Winther Andersen†, Hartwig R. Siebner†,
Tim B. Dyrby*†, Kristoffer H. Madsen*†, Mikkel N. Schmidt*, and Morten Mørup*
*Department of Applied Mathematics and Computer Science, Technical University of Denmark, Lyngby, Denmark
†Danish Research Centre for Magnetic Resonance, Copenhagen University Hospital Hvidovre, Hvidovre, Denmark

***Abstract*—Modern MRI techniques provide non-invasive high-resolution images from which networks of whole-brain functional and structural connectivity can be derived. Though it is commonly believed that structure and function emerge from the same modular organization of the brain, it remains elusive how and to what extend the two modalities are related. Assuming that structure and function express the same fundamental organization of the brain, we hypothesize that data-driven parcellation based jointly on functional and structural connectivity should better define processing units than parcellation based on each modality separately. Whereas previous studies have primarily focused on investigating how one modality can characterize the other, we investigate how parcellations derived using both modalities characterize the individual modalities, and compare this to parcellations derived from the modalities individually. We use a stochastic block-model as a data-driven approach for clustering high-resolution whole-brain connectivity networks, assuming identical parcellation across modalities and independent connectivity structure between parcels. We evaluate the parcellations by their ability to predict structure in held-out test subjects' functional and structural data: In predicting functional connectivity, we show a substantial improvement by including structural information, even when the functional networks are averaged across 50 subjects; however, in predicting structural connectivity, we show no benefit from including functional information. We attribute this asymmetry to a higher level of noise in the functional networks, possibly caused by the dynamic nature of neuronal function in comparison to the static structural modality. Our results reveal that the modular structures expressed by functional and structural networks are consistent, but the connectivity between parcels is substantially different.**

I. Introduction

A prominent way to view the organization of the brain on a macro scale is to consider two fundamental aspects: While the cortex is *segregated* into specialized neuronal regions, the cognitive functions emerges from *integration* of these regions by coordinated activation [30]. Both in terms of its structural organization and functional activity the brain can be studied as a network, allowing network science to provide the statistical foundation and methodology for investigating and quantifying the organization of brain connectivity networks [8].

One approach of quantifying the latent structure in connectivity networks is to partition the nodes into groups that share a similar connectivity pattern within the network. The stochastic block model (SBM) [21] is a data-driven Bayesian clustering approach, which coupled with Markov Chain Monte Carlo (MCMC) sampling has proven a valid tool for clustering and investigating structure in complex networks [36], [25]. Notably, a non-parametric SBM modeling framework has previously been used [4] for the joint modeling of structural and functional connectivity based on low resolution networks of 116 nodes defined by the AAL atlas [32] with the ability to impose shared and individual segregated units of the two modalities.

Magnetic Resonance Imaging (MRI) techniques provides non-invasive means from which functional and structural connectivity networks can be constructed. Structural connectivity can be derived from diffusion MRI [14] by tracking white matter streamlines across the cortex such that structural networks are obtained based on the anatomy of the brain. Functional MRI captures images of functional whole brain connectivity by indirectly measuring the time-dependent neural activity within small regions of the brain (i.e., voxels) by monitoring the blood oxygenation level dependent (BOLD) response [22]. Networks of functional connectivity can be obtained, for instance as mapped by the correlated activation of brain regions [8].

The extend to which the structural and functional organization is related and how to quantify the relationship remains a challenging and prominent area of active research. Previous studies suggest that there are relations between the two modalities. This can for instance be anticipated by the network properties, such as functional connectivity networks exhibiting various small-world attributes [1], which evolutionary could be reflected by economical efficient structure [8]. Another hint comes from the fact that various neurological disorders have shown to cause alterations in both functional and structural connectivity [11], [31], but to what extend any relation between functional and structural connectivity effects brain disease needs further investigation [34]. On the whole-brain scale previous studies suggest that the functional connectivity to some extend emerges from the structural organization [5], [29], [15]. Data from both modalities are often modelled individually but used to enhance each other, for example by using one modality to define regions of interest that are afterwards examined in the other modality, or by predicting data from one modality from information obtained from the other modality [5].

If structure and function express the same fundamental organization of the brain they should both inform about the brain's organization into segregated processing units. We thus hypothesize that data-driven parcellations based on joint modeling of functional and structural connectivity data should better define these processing units than modeling each modality separately with limited data. To investigate this, we use the SBM which allows to infer a single parcellation based on multiple networks and provides sound statistical evaluation of the predictive performance of the inferred parcellations. We exploit this to jointly model functional and structural data and compare it with modelling data from the modalities individually. We further compare the results of joint modelling with

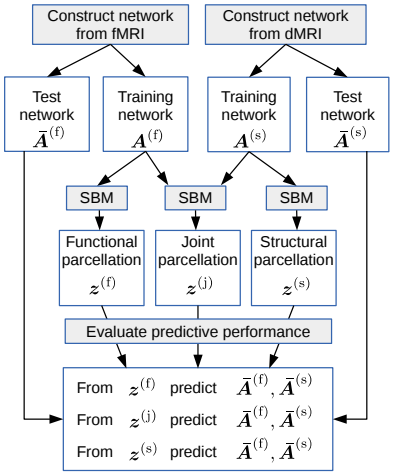


Fig. 1. Concept for the modelling approach. Networks are obtained from both fMRI and dMRI neuroimages. The networks are split into training and test data. The data-driven stochastic block model (SBM) is used to infer parcellations based on functional training data, structural training data, and jointly based on both functional and structural training data. Using the inferred model parameters and training networks the predictive performance on test data is computed.

the comprehensive HCP_MMP1.0 atlas which is constructed using multiple modalities as well as neuroanatomy [12].

II. Data and modelling framework

Figure 1 illustrates the modelling framework. Based on fMRI and dMRI images, networks of structural and functional connectivity are created for a number of subjects. The networks are split into training and test data, such that parcellations are inferred by SBM from the training data, while test networks from unseen subjects are used to asses the quality of the parcellations as quantified by the predictive performance. SBM allows us to obtain a parcellation learned solely from either the structural or functional test data as well as learning a single parcellation jointly derived from data of both modalities. Using the inferred model parameters and training networks the predictive performance on the test networks is compared to evaluate the influence of jointly modelling both modalities.

A. Data

Networks of brain connectivity were obtained using independent high-resolution data from the Human Connectome Project (HCP) [33], [19], [10], [26], [35] database. Ignoring the sub-cortical information the networks contained 59 412 vertices. We split the subjects into populations, such that we obtained 5 non-overlapping groups of subjects for each of the training population sizes of 1, 2, 10 and 50 subjects. For each group we created a single functional and structural training network based on the group average. Similarly we constructed test networks based on the group average of 50 held-out subjects. The fMRI networks were estimated from the preprocessed and structurally denoised ICA-FIX cleaned version of the resting state fMRI data, for further reference see [27], [16], [24]. We formed the networks by averaging the Pearson correlation matrix estimated from the two sessions using both the left-right and right-left phase encoding directions for each subject (i.e., averaging four correlograms per subject each estimated from 1200 time frames). The structural connectivity networks were derived from the dMRI data preprocessed using the HCP pipeline [13]. The fiber orientation estimation was done using FSL's BedpostX for multi-shell data [18] and the networks were constructed by performing probabilistic tractography using FSL's Probtrackx2 [7], [6] run in "matrix3" mode. 1000 streamlines were initiated in each white matter voxel and kept if it reached two vertices of the white matter surface, resulting in weighted graphs of streamline counts between vertices. The adjacency matrices were added and binarized by thresholding the graph at 1% density keeping only the strongest links.

The HCP_MMP1.0 atlas [12] is based on multi-modal MRI data from HCP and describes a total of 360 parcels split equally across both hemispheres. It was created in a combined data-driven and manual approach to obtain a single parcellation of cortical regions, based on multiple neurobiological properties including both functional information and brain anatomy obtained from 210 healthy subjects. It has previously been shown to be very efficient for predicting structural connectivity networks [3].

B. The stochastic block model

The stochastic block model (SBM) [21] partitions network nodes into clusters with similar connectivity patterns. For modelling binary networks, the model can be defined by the following generative process, where m is used to index modality:

$$\text{Links in network:} \quad \boldsymbol{A}_{ij}^{(m)} \sim \text{Bernoulli}(\eta_{z_i z_j}^{(m)}), \tag{1}$$

$$\text{Cluster-link densities:} \quad \eta_{lh}^{(m)} \sim \text{Beta}(\beta^+, \beta^-), \tag{2}$$

$$\text{Clustering:} \quad \boldsymbol{z} \sim \text{Categorical}(\boldsymbol{\pi}), \tag{3}$$

$$\text{Cluster proportions:} \quad \boldsymbol{\pi} \sim \text{Dirichlet}(\alpha). \tag{4}$$

The probability of observing a link between two nodes i and j in the network are considered generated according to a Bernoulli distribution, only depending on the probability of observing links between the clusters z_i and z_j that the nodes belong to. The probability of observing links between two clusters is considered independent given the assignment to clusters and follows a Beta distribution. Finally, the nodes are partitioned into K clusters based on the Dirichlet distribution. Due to the conjugacy between the Dirichlet and Cateogrical distribution, $\boldsymbol{\pi}$ can be analytically marginalized (see [25] for details). By imposing an equal concentration parameter for all clusters the following effective prior for the clustering can be obtained:

$$p(\boldsymbol{z}|\alpha) = \frac{\Gamma(\alpha)}{\Gamma(\alpha + N)} \prod_{l=1}^{K} \frac{\Gamma(\frac{\alpha}{K} + n_k)}{\Gamma(\frac{\alpha}{K})}, \tag{5}$$

where N is the number of nodes, n_k is the number of nodes in cluster k, and $\Gamma(x)$ is the gamma function. Notably, we use the SBM to obtain a *single* parcellation based on either a network from one modality (either functional or structural) or using both modalities. Let $\boldsymbol{A}$ represent the set of M networks, containing either $M = 1$ or $M = 2$ modalities. The beta prior is conjugate to the Bernoulli likelihood, which allows us to obtain the following joint distribution as $\boldsymbol{\eta}$ can be analytically

integrated out:

$$P(\boldsymbol{A}, \boldsymbol{z}|\beta^+, \beta^-, \alpha) = P(\boldsymbol{z}|\alpha) \prod_m^M \prod_{l<h} \frac{B(N_{lh}^{(m)+} + \beta^+, N_{lh}^{(m)-} + \beta^-)}{B(\beta^+, \beta^-)}, \quad (6)$$

where $N_{lh}^{(m)+}$ and $N_{lh}^{(m)-}$ respectively represent the number of links and non-links between cluster l and h according to network $\boldsymbol{A}^{(m)}$, while $B(a,b) = \frac{\Gamma(a)\Gamma(b)}{\Gamma(a+k)}$ is the beta function.

C. Inference procedure

We infer the model parameters using a Markov Chain Monte Carlo (MCMC) procedure. The parcellation is inferred by Gibbs sampling, where the assignment z_i for each node i in turn is processed, based on the posterior distribution for the assignment of i to each of the K clusters ℓ. Using Bayes' theorem this can be obtained from equation 6, where $\boldsymbol{z}^{\setminus i}$ is the cluster assignments for all nodes ignoring node i:

$$P(z_i = \ell|\boldsymbol{z}^{\setminus i}, \beta^+, \beta^-, \alpha) = \frac{P(\boldsymbol{A}, \boldsymbol{z}^{\setminus i}, z_i = \ell|\beta^+, \beta^-, \alpha)}{\sum_{h=1}^{K} P(\boldsymbol{A}, \boldsymbol{z}^{\setminus i}, z_i = h|\beta^+, \beta^-, \alpha)}. \quad (7)$$

For inferring the hyper-parameters we use a simple Metropolis-Hastings procedure, where new proposals are drawn from a Gaussian distribution centred at the current parameter value with variance 1.

For all experiments the model parameters are inferred by sampling 100 iterations of the following sampling procedure: $\boldsymbol{z}$ is updated by one complete Gibbs sweep over all nodes followed by 1000 MH-proposals for updating each hyper-parameter β^+, β^-, α. Due to the size of the networks and behaviour of the Gibbs sampler, it is not computationally feasible to reach convergence [2]. We hence treat the last sampled state as the inferred parameters. All experiments are performed with $K = 360$ clusters which limits SBM to the same complexity as the HCP_MMP1.0 atlas.

D. Predictive performance by AUC

To assess and compare the quality of parcellations we use the predictive framework established in [3]. The quality of a parcellation is evaluated by how well it can be used to predict unseen held-out networks. We quantify this performance by the area under curve (AUC) of the Receiver Operator Characteristics curve (ROC) [9], scored by the expected link probability between clusters:

$$\langle \eta_{z_i z_j}^{(m)} \rangle = \frac{N_{z_i z_j}^{(m)+} + \beta^+}{N_{z_i z_j}^{(m)+} + N_{z_i z_j}^{(m)-} + \beta^+ + \beta^-}. \quad (8)$$

As baseline we also compute AUC by using the HCP_ MMP1.0 atlas instead of the inferred parcellations. In this case we let $\beta^+ = \beta^- = 0$ and note that the AUC values hence cannot be used to directly compare the quality of parcellations between the atlas and SBM as the regularization induced by these hyper-parameters in SBM may benefit prediction. As baseline we also evaluate how well the raw training networks can predict the test network.

E. Parcellation comparison by Mutual Information

The similarity of different parcellations can be quantified using Mutual Information (MI). This constitutes a permutation invariant measure for the shared clustering information between two parcellations $\boldsymbol{z}$ and $\boldsymbol{z'}$ given by:

$$\mathrm{MI}(\boldsymbol{z}, \boldsymbol{z'}) = \sum_{cc'} P(c, c') \log \left(\frac{P(c, c')}{P(c)P(c')} \right), \quad (9)$$

where $P(c)$ is the probability of observing a node in cluster c while $P(c, c')$ is the probability of jointly observing a node in cluster c in $\boldsymbol{z}$ and a node in cluster c' in $\boldsymbol{z'}$. We use the normalized mutual information (NMI) to get a value between zero and one:

$$\mathrm{NMI}(\boldsymbol{z}, \boldsymbol{z'}) = \frac{2\ \mathrm{MI}(\boldsymbol{z}, \boldsymbol{z'})}{\mathrm{MI}(\boldsymbol{z}, \boldsymbol{z}) + \mathrm{MI}(\boldsymbol{z'}, \boldsymbol{z'})}, \quad (10)$$

such that a value of one indicates that the parcellations are identical.

III. Results

Figure 2 and 3 respectively show the results of predicting the functional and structural test networks based on the following:

- Parcellations inferred from SBM trained on data for a single modality.
- Parcellations inferred from SBM trained jointly on both modalities.
- The fixed multi-modal HCP_ MMP1.0 atlas used to predict test data based on the training networks with the same modality.
- The raw graph match between the test networks and training network with the same modality.

AUC scores are individually computed and averaged for the 5 training networks for each population size of 1,2,10 and 50 subjects, when predicting on the same 50 subject population held-out network of respectively functional and structural connectivity.

For both modalities the AUC score improves for larger population sizes while the standard deviation decreases drastically. The predictive performance for the atlas also improves for larger population sizes but the model based approaches presents more pronounced improvements. Considering the predictions based on the fixed HCP_ MMP1.0 atlas parcellations as well as predictions based on the raw graphs we observe that the structural networks are more similar than functional networks. We attribute this to the structural connectivity data being more static and less noisy than functional data.

Figure 2 shows that the model parcellations provide a better prediction of functional connectivity when SBM is trained on both modalites compared to just functional connectivity data. Though this effect is true for all population sizes this effect is reduced when the noise is reduced by including larger populations. Interestingly for few subjects, the structural parcellation predicts the functional data better than both a functional and joint parcellation. This effect diminishes for larger populations, and at 50 subjects the structural parcellation allows for predictions on par with the functional parcellation while the joint is slightly better. Figure 3 shows that predicting structural connectivity does not benefit from joint modelling as

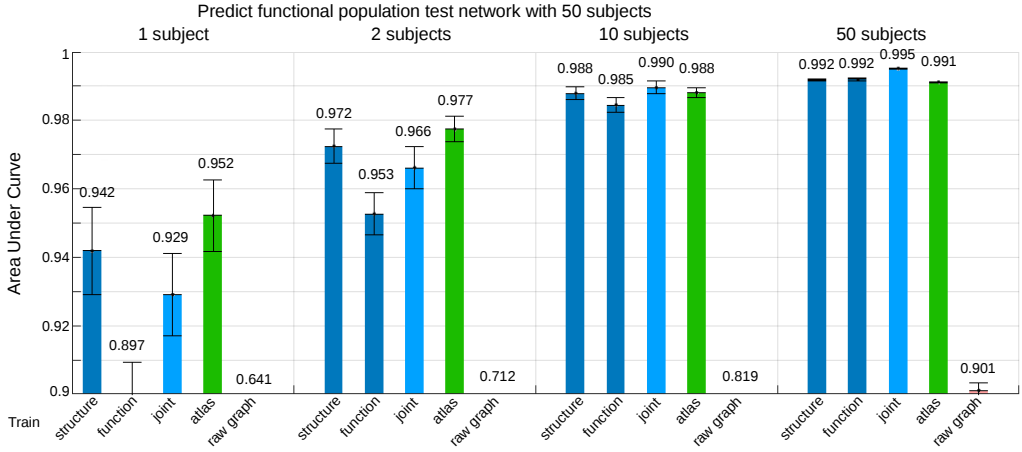


Fig. 2. AUC when prediction from the functional training networks to the single functional network for the test population of 50 subjects. For each training population size, the bars show the average AUC score as obtained by using different parcellations: (1) SBM on the structural training networks, (2) SBM on the functional training network, (3) SBM on both structural and functional training networks, (4) the atlas parcellation used with the functional training networks, and (5) the raw graph match between the functional training and test networks. For each population size five training networks were utilized. The mean value is shown above each bar while the whiskers indicate the standard deviation of the mean ($\pm \mathrm{std}/\sqrt{5}$).

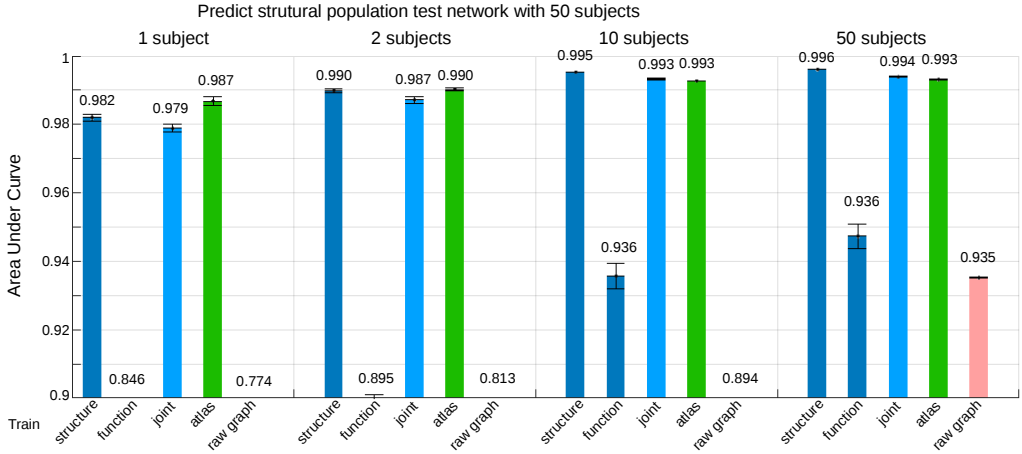


Fig. 3. AUC when prediction from the structural training networks to the structural network for the test population of 50 subjects. For each training population size, the bars show the average AUC score as obtained by using different parcellations: (1) SBM on the structural training networks, (2) SBM on the functional training networks, (3) SBM on both structural and functional training networks, (4) the atlas parcellation used with the structural training networks, and (5) the raw graph match between the structural training and test networks. For each population size five training networks were utilized. The mean value is shown above each bar while the whiskers indicate the standard deviation of the mean ($\pm \mathrm{std}/\sqrt{5}$).

the model parcellations inferred from structural networks alone gives a higher AUC for all considered training population sizes.

Figure 4 shows the Normalized Mutual Information for the different parcellation strategies. The figure shows a high NMI between the atlas and parcellations inferred from both structure and joint modelling. Parcellations based on structural connectivity is in better agreement with the atlas than parcellations based on functional parcellations. For single subject networks the NMI is higher between the atlas and SBM parcellations than between individual SBM parcellations for both structure and function. When more data is included in the modelling the inferred parcellations becomes more in agreement for both modalities. Especially between parcellations inferred from function the NMI increases rapidly for larger population sizes. This effect we attribute to the de-noising of the functional data achieved when averaged over more subjects.

Figure 5 shows the estimated link density using the atlas parcellation or a single SBM parcellation for joint modelling. These results are based on the same single training network of structural and functional connectivity for 50 subjects and the two test networks also averaged over 50 subjects. The estimated link densities are computed using equation 8. For both the model and atlas parcellation the figure illustrates the connectivity profile between the structure and function training network, and between the test and training networks for the same modalities. The figure shows that the functional and structural connectivity profiles are very different. For both SBM and the atlas the extracted structural connectivity profiles are in high agreement while the functional connectivity data is less in agreement in particular for the SBM parcellation.

IV. Discussion

We have demonstrated that the stochastic block model (SBM) intuitively can be used to jointly infer a parcellation integrating functional and structural networks of brain connectivity. Notably, while the segregated regions are the same the framework proposed assumes the functional and structural connectivity structure to be independent.

For both modalities the AUC between raw graphs improves drastically the larger the population size. This shows that networks for larger populations ought to have an improved signal-to-noise ratio and are thus more consistent. We, however, still observe that the joint modelling provides a better predictive performance for fMRI data. This provides evidence for the two modalities indeed are organized in terms of shared segregated processing units and that fusion of modalities is beneficial when dealing with noisy data such as fMRI.

The effect of de-noising using larger population sizes is also evident from the NMI of the inferred parcellations. For single subjects the NMI is higher between any model inferred parcellation and the atlas than between any two model parcellations. As expected the NMI increases for larger population sizes, and model parcellations becomes more similar than compared to the atlas. The effect is most clearly visible for functional networks. For a single subject, the NMI between structure and function is actually higher than the NMI between any two parcellations that are both inferred from function. This emphasises that noisy functional data benefit by the inclusion of structural information. Furthermore the NMI shows that particularly the structural parcellations to a great extend comply with the multimodal HCP_MMP1.0 atlas as also observed in [3] and slightly better than the parcellations based on joint modeling.

Similar to [17] we observe siginificant correspondence between the extracted functional and structural connectivity profiles. However, despite the fundamental functional and structural processing units estimated as being the same we observe that the functional and structural connectivity profiles substantially differ when compared to the connectivity variability within each modality across independent population data. Thus, while both modalities provide information regarding the brains organization into segregated processing units there are important differences in terms of the strenghts of functional and structural connectivity beyond the variability observed within each modality. Thus, while the segregated units may be shared our results demonstrate that the data generally violates assumptions of consistency of structural and functional connectivity profiles. However for joint modelling these connectivity profiles seem to be more consistent than for the atlas, as evident from a slightly higher correlation between function and structure for the SBM parcellation.

In the present study we considered the perhaps most simple approach to extracting functional connectivity based on zero lag pearson correlation [8], [28], [23]. Notably, it is unclear how functional connectivity is best quantified and several approaches exists including mutual information [8], [20], [28], wavelet correlation [1], lagged correlation and partial correlation, as well as approaches quantifying directionality, see also [28], [23] for reviews. We presently considered only positive correlation while negative functional correlations arguable also relate to structure. We further notice that the examined HCP fMRI data has a high temopral and spatial resolution, which might give an poorer signal-to-noise ratio than other protocols. Furthermore, we arbitrarily thresholded the networks at 1 percent density. Future work should thus investigate the influence of network construction.

We find that multimodal modeling is beneficial in particular when facing noisy data. Importantly, or modeling assumed network nodes were correctly aligned to correspond to the same structure in the brain across subjects. Whereas misalignment can be considered a source of noise, systematic modality specific biases will in general reduce potential structure-function relationships. Our finding of large differences in functional and structural connectivity profiles may thus be caused by modality specific biases including biases in the functional and structural network construction. We presently considered joint modeling of structural and functional connectivity data, however, the proposed framework naturally extends to general multimodal modeling including additional modalities.

Acknowledgment

The MRI data used in this work were obtained from the MGH-USC Human Connectome Project (HCP) database (https://ida.loni.usc.edu/login.jsp) in the "500 subjects" release. The HCP project (Principal Investigators: Bruce Rosen, M.D., Ph.D., Martinos Center at Massachusetts General Hospital; Arthur W. Toga, Ph.D., University of California, Los Angeles, Van J. Weeden, MD, Martinos Center at Massachusetts General Hospital) is supported by the National Institute of Dental and Craniofacial Research (NIDCR), the National Institute of Mental Health (NIMH) and the National Institute of Neurological Disorders and Stroke (NINDS). Collectively, the HCP is the result of efforts of co-investigators from the University of California, Los Angeles, Martinos Center for Biomedical Imaging at Massachusetts General Hospital (MGH), Washington University, and the University of Minnesota.

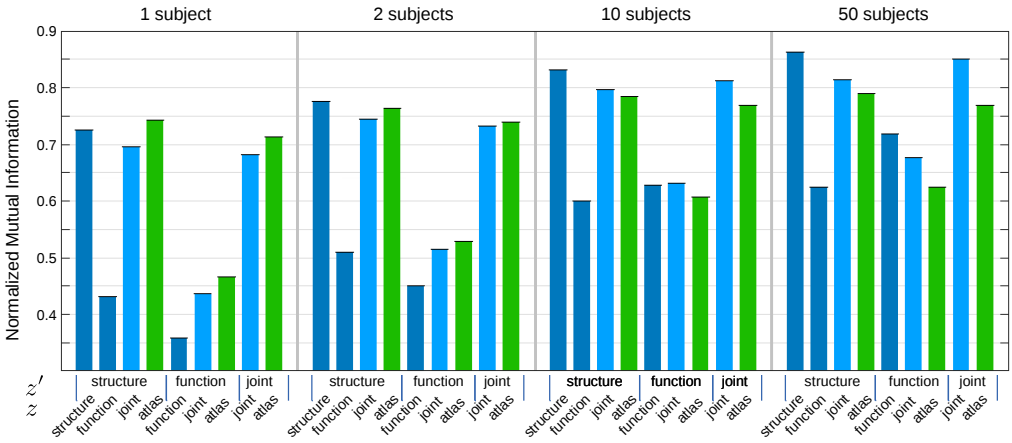


Fig. 4. Normalized Mutual Information, NMI(z, z'), between model parcellations inferred from data of the same modality as well as between inferred parcellations and the HCP_MMP1.0 atlas. The NMI values are averaged over five comparisons for each approach (jointly modelling both modalities and individually modelling structure and function) for each population size of 1, 2, 10 and 50 subjects. The results does not include NMI between different parcellations inferred from the same group of subjects. For all bars the standard error of the mean (std $\pm\sqrt{5}$) is less than 0.011.

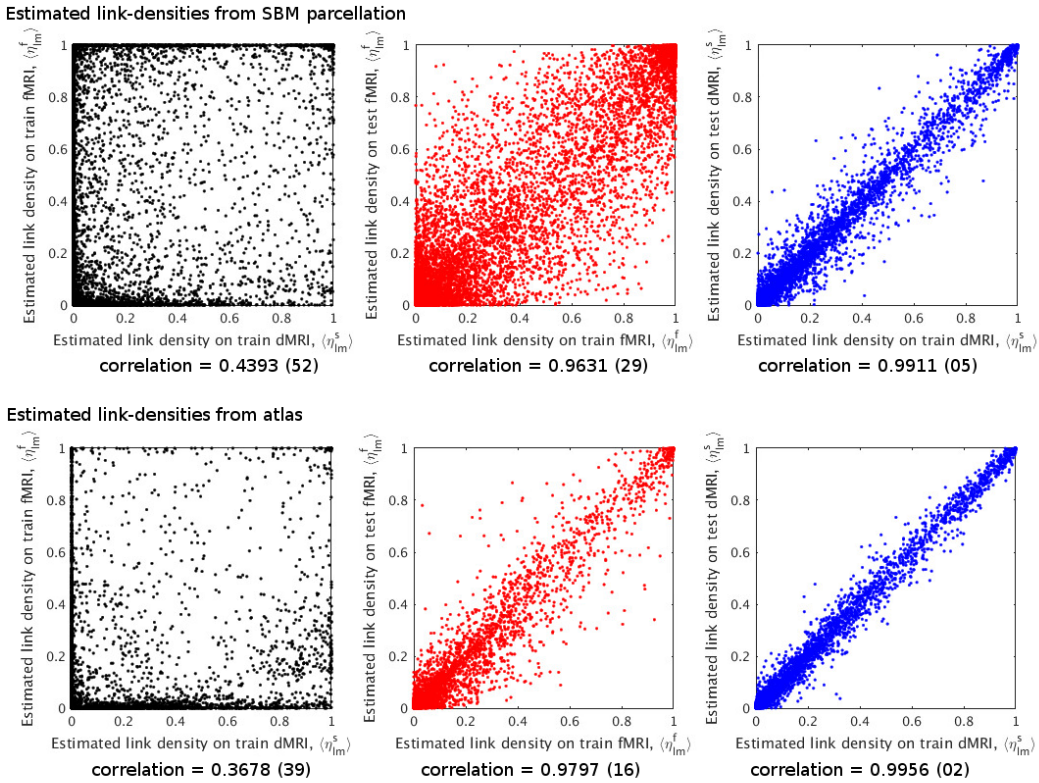


Fig. 5. Estimated link-densities for SBM with joint modelling of both modalities for a population of 50 subjects and using the atlas parcellation. The figure illustrates the connectivity profiles for both functional and structural networks. It compares the estimated link density between the structural and functional training networks for the same 50 subject population, and between the training and test networks for respectively structural and functional connectivity. The plots shows for a single parcellation, while the correlations below the plots are the mean for all five training networks, all correlations are significant ($p = 0$) by the $360 \times (360 + 1)/2 = 64.980$ elements. The standard deviation of the mean on the last digits is shown in parentheses.

This project was funded by the Lundbeck Foundation, grant nr. R105-9813.

REFERENCES

[1] Sophie Achard, Raymond Salvador, Brandon Whitcher, John Suckling, and ED Bullmore. A resilient, low-frequency, small-world human brain functional network with highly connected association cortical hubs. *Journal of Neuroscience*, 26(1):63–72, 2006.

[2] Kristoffer Jon Albers, Andreas Leon Aagard Moth, Morten Morup, and Mikkel N Schmidt. Large scale inference in the infinite relational model: Gibbs sampling is not enough. In *Machine Learning for Signal Processing (MLSP), 2013 IEEE International Workshop on*, pages 1–6. IEEE, 2013.

[3] Karen S Ambrosen, Kristoffer Jon Albers, Matthew G Libtrot, Tim B Dyrby, Mikkel N Schmidt, and Morten Mørup. Predictive validation of human brain parcellation. *submitted, under review*, 2017.

[4] Kasper Winther Andersen, Tue Herlau, Morten Mørup, Mikkel Nørgaard Schmidt, Kristoffer H Madsen, Mark Lyksborg, Tim B Dyrby, Hartwig R Siebner, and Lars Kai Hansen. Joint modelling of structural and functional brain networks. In *2nd NIPS Workshop on Machine Learning and Interpretation in NeuroImaging (MLINI 2012)*, 2012.

[5] Cassiano O Becker, Sergio Pequito, George J Pappas, Michael B Miller, Scott T Grafton, Danielle S Bassett, and Victor M Preciado. Accurately predicting functional connectivity from diffusion imaging. *arXiv preprint arXiv:1512.02602*, 2015.

[6] TEJ Behrens, H Johansen Berg, Saad Jbabdi, MFS Rushworth, and MW Woolrich. Probabilistic diffusion tractography with multiple fibre orientations: What can we gain? *Neuroimage*, 34(1):144–155, 2007.

[7] TEJ Behrens, MW Woolrich, M Jenkinson, H Johansen-Berg, RG Nunes, S Clare, PM Matthews, JM Brady, and SM Smith. Characterization and propagation of uncertainty in diffusion-weighted mr imaging. *Magnetic resonance in medicine*, 50(5):1077–1088, 2003.

[8] Ed Bullmore and Olaf Sporns. Complex brain networks: graph theoretical analysis of structural and functional systems. *Nature Reviews Neuroscience*, 10(3):186–198, 2009.

[9] Aaron Clauset, Cristopher Moore, and Mark EJ Newman. Hierarchical structure and the prediction of missing links in networks. *Nature*, 453(7191):98–101, 2008.

[10] David A Feinberg, Steen Moeller, Stephen M Smith, Edward Auerbach, Sudhir Ramanna, Matt F Glasser, Karla L Miller, Kamil Ugurbil, and Essa Yacoub. Multiplexed echo planar imaging for sub-second whole brain fmri and fast diffusion imaging. *PloS one*, 5(12):e15710, 2010.

[11] Alex Fornito and EdwardT Bullmore. Connectomic intermediate phenotypes for psychiatric disorders. *Magnetic resonance imaging of disturbed brain connectivity in psychiatric illness*, page 6, 2012.

[12] Matthew F Glasser, Timothy S Coalson, Emma C Robinson, Carl D Hacker, John Harwell, Essa Yacoub, Kamil Ugurbil, Jesper Andersson, Christian F Beckmann, Mark Jenkinson, et al. A multi-modal parcellation of human cerebral cortex. *Nature*, 2016.

[13] Matthew F Glasser, Stamatios N Sotiropoulos, J Anthony Wilson, Timothy S Coalson, Bruce Fischl, Jesper L Andersson, Junqian Xu, Saad Jbabdi, Matthew Webster, Jonathan R Polimeni, et al. The minimal preprocessing pipelines for the human connectome project. *Neuroimage*, 80:105–124, 2013.

[14] Gaolang Gong, Yong He, Luis Concha, Catherine Lebel, Donald W Gross, Alan C Evans, and Christian Beaulieu. Mapping anatomical connectivity patterns of human cerebral cortex using in vivo diffusion tensor imaging tractography. *Cerebral cortex*, 19(3):524–536, 2009.

[15] Michael D Greicius, Kaustubh Supekar, Vinod Menon, and Robert F Dougherty. Resting-state functional connectivity reflects structural connectivity in the default mode network. *Cerebral cortex*, 19(1):72–78, 2009.

[16] Ludovica Griffanti, Gholamreza Salimi-Khorshidi, Christian F. Beckmann, Edward J. Auerbach, Gwenaëlle Douaud, Claire E. Sexton, Enik Zsoldos, Klaus P. Ebmeier, Nicola Filippini, Clare E. Mackay, Steen Moeller, Junqian Xu, Essa Yacoub, Giuseppe Baselli, Kamil Ugurbil, Karla L. Miller, and Stephen M. Smith. ICA-based artefact removal and accelerated fMRI acquisition for improved resting state network imaging. *NeuroImage*, 95:232–247, 2014.

[17] CJ Honey, O Sporns, Leila Cammoun, Xavier Gigandet, Jean-Philippe Thiran, Reto Meuli, and Patric Hagmann. Predicting human resting-state functional connectivity from structural connectivity. *Proceedings of the National Academy of Sciences*, 106(6):2035–2040, 2009.

[18] Saad Jbabdi, Stamatios N Sotiropoulos, Alexander M Savio, Manuel Graña, and Timothy EJ Behrens. Model-based analysis of multishell diffusion mr data for tractography: How to get over fitting problems. *Magnetic Resonance in Medicine*, 68(6):1846–1855, 2012.

[19] Steen Moeller, Essa Yacoub, Cheryl A Olman, Edward Auerbach, John Strupp, Noam Harel, and Kâmil Uğurbil. Multiband multislice ge-epi at 7 tesla, with 16-fold acceleration using partial parallel imaging with application to high spatial and temporal whole-brain fmri. *Magnetic Resonance in Medicine*, 63(5):1144–1153, 2010.

[20] Morten Mørup, Kristoffer Madsen, Anne-marie Dogonowski, Hartwig Siebner, and Lars K Hansen. Infinite relational modeling of functional connectivity in resting state fmri. In *Advances in neural information processing systems*, pages 1750–1758, 2010.

[21] Krzysztof Nowicki and Tom A B Snijders. Estimation and prediction for stochastic blockstructures. *Journal of the American Statistical Association*, 96(455):1077–1087, 2001.

[22] Seiji Ogawa, Tso-Ming Lee, Alan R Kay, and David W Tank. Brain magnetic resonance imaging with contrast dependent on blood oxygenation. *Proceedings of the National Academy of Sciences*, 87(24):9868–9872, 1990.

[23] Jonas Richiardi, Sophie Achard, Horst Bunke, and Dimitri Van De Ville. Machine learning with brain graphs: predictive modeling approaches for functional imaging in systems neuroscience. *IEEE Signal Processing Magazine*, 30(3):58–70, 2013.

[24] Gholamreza Salimi-Khorshidi, Gwenaëlle Douaud, Christian F. Beckmann, Matthew F. Glasser, Ludovica Griffanti, and Stephen M. Smith. Automatic denoising of functional MRI data: Combining independent component analysis and hierarchical fusion of classifiers. *NeuroImage*, 90:449–468, 2014.

[25] Mikkel N Schmidt and Morten Morup. Nonparametric bayesian modeling of complex networks: An introduction. *Signal Processing Magazine, IEEE*, 30(3):110–128, 2013.

[26] Kawin Setsompop, Borjan A Gagoski, Jonathan R Polimeni, Thomas Witzel, Van J Wedeen, and Lawrence L Wald. Blipped-controlled aliasing in parallel imaging for simultaneous multislice echo planar imaging with reduced g-factor penalty. *Magnetic Resonance in Medicine*, 67(5):1210–1224, 2012.

[27] Stephen M. Smith, Christian F. Beckmann, Jesper Andersson, Edward J. Auerbach, Janine Bijsterbosch, Gwenaëlle Douaud, Eugene Duff, David A. Feinberg, Ludovica Griffanti, Michael P. Harms, Michael Kelly, Timothy Laumann, Karla L. Miller, Steen Moeller, Steve Petersen, Jonathan Power, Gholamreza Salimi-Khorshidi, Abraham Z. Snyder, An T. Vu, Mark W. Woolrich, Junqian Xu, Essa Yacoub, Kamil Uurbil, David C. Van Essen, and Matthew F. Glasser. Resting-state fMRI in the Human Connectome Project. *NeuroImage*, 80:144–168, 2013.

[28] Stephen M Smith, Karla L Miller, Gholamreza Salimi-Khorshidi, Matthew Webster, Christian F Beckmann, Thomas E Nichols, Joseph D Ramsey, and Mark W Woolrich. Network modelling methods for fmri. *Neuroimage*, 54(2):875–891, 2011.

[29] Olaf Sporns. Contributions and challenges for network models in cognitive neuroscience. *Nature neuroscience*, 17(5):652–660, 2014.

[30] Giulio Tononi, Olaf Sporns, and Gerald M Edelman. A measure for brain complexity: relating functional segregation and integration in the nervous system. *Proceedings of the National Academy of Sciences*, 91(11):5033–5037, 1994.

[31] Heike Tost, Edda Bilek, and Andreas Meyer-Lindenberg. Brain connectivity in psychiatric imaging genetics. *Neuroimage*, 62(4):2250–2260, 2012.

[32] N. Tzourio-Mazoyer, B. Landeau, D. Papathanassiou, F. Crivello, O. Etard, N. Delcroix, B. Mazoyer, and M. Joliot. Automated anatomical labeling of activations in {SPM} using a macroscopic anatomical parcellation of the {MNI} {MRI} single-subject brain. *NeuroImage*, 15(1):273 – 289, 2002.

[33] David C Van Essen, Stephen M Smith, Deanna M Barch, Timothy EJ Behrens, Essa Yacoub, Kamil Ugurbil, WU-Minn HCP Consortium, et al. The wu-minn human connectome project: an overview. *Neuroimage*, 80:62–79, 2013.

[34] Sandro Vega Pons, Emanuele Olivetti, Paolo Avesani, Luca Dodero, Alessandro Gozzi, and Angelo Bifone. Differential effects of brain

disorders on structural and functional connectivity. *Frontiers in Neuroscience*, 10, 2016.

[35] J Xu, S Moeller, J Strupp, E Auerbach, L Chen, DA Feinberg, K Ugurbil, and E Yacoub. Highly accelerated whole brain imaging using aligned-blipped-controlled-aliasing multiband epi. In *Proceedings of the 20th Annual Meeting of ISMRM*, volume 2306, 2012.

[36] Shenghuo Zhu, Kai Yu, and Yihong Gong. Stochastic relational models for large-scale dyadic data using mcmc. In *Advances in Neural Information Processing Systems*, pages 1993–2000, 2009.

Appendix G

Stochastic Blockmodels

For the stochastic blockmodels, this appendix presents how to derive the expression for the posterior distribution $P(\boldsymbol{A}, \boldsymbol{z})$ by marginalizing the cluster-link-probabilities. Depending on the type of links in the network different model definitions are preferable, with different distributions utilized for the likelihood and conjugate prior.

Link types	Likelihood	Conjugate prior
Binary	*Bernoulli*	*Beta*
Counts	*Poisson*	*Gamma*
Categories	*Categorical*	*Dirichlet*
Continuous	*Normal*	*Normal-Inverse-Gamma*

G.1 Bernoulli likelihood and Beta prior

For a network with unweighted (binary) links, the link probabilities can intuitively be modelled using the Bernoulli distribution. With the Beta distribution acting as conjugate prior, the generative model becomes:

Links	$A_{ij} \sim Bernoulli(\eta_{z_i,z_j})$
Interactions	$\eta_{lm} \sim Beta(\beta^+, \beta^-)$
Groups (infinite)	$\boldsymbol{z} \sim CRP(\alpha)$
Groups (finite)	$\boldsymbol{z} \sim Dirichlet\text{-}Categorical(\alpha)$

For the interaction between two groups l and m, the Beta distribution gives:

$$Beta(\eta_{lm}|\beta^+, \beta^-) = \frac{\Gamma(\beta^+ + \beta^-)}{\Gamma(\beta^+)\Gamma(\beta^-)} \eta_{lm}^{\beta^+ - 1}(1 - \eta_{lm})^{\beta^- - 1}$$

For all pairwise independent interactions, this gives:

$$P(\boldsymbol{\eta}|\beta^+, \beta^-) = \prod_{l \leq m} \frac{\Gamma(\beta^+ + \beta^-)}{\Gamma(\beta^+)\Gamma(\beta^-)} \eta_{lm}^{\beta^+ - 1}(1 - \eta_{lm})^{\beta^- - 1} \tag{G.1}$$

For the link between two nodes i and j, the Bernoulli distribution gives:

$$Bernoulli(A_{ij}|\eta_{z_i z_j}) = \eta_{z_i z_j}^{A_{ij}}(1 - \eta_{z_i z_j})^{1 - A_{ij}}$$

For all links $\boldsymbol{A}$ this gives:

$$P(\boldsymbol{A}|\boldsymbol{\eta}) = \prod_{l \leq m} \eta_{lm}^{N_{lm}^+}(1 - \eta_{z_i z_j})^{N_{lm}^-}, \tag{G.2}$$

where N^+ is the number of links between group l and m while N_{lm}^- is the number of possible yet not observed links between group l and m.

By joining G.1 and G.2 we obtain the following joint distribution:

$$\begin{aligned} P(\boldsymbol{A}, \boldsymbol{z}, \boldsymbol{\eta}|\alpha, \beta^+, \beta^-) &= P(\boldsymbol{z}|\alpha) \times P(\boldsymbol{A}|\boldsymbol{\eta}) \times P(\boldsymbol{\eta}|\beta^+, \beta^-) \\ &= P(\boldsymbol{z}|\alpha) \times \prod_{l \leq m} \left(\frac{\Gamma(\beta^+ + \beta^-)}{\Gamma(\beta^+)\Gamma(\beta^-)} \eta_{lm}^{\beta^+ N_{lm}^+ - 1}(1 - \eta_{lm})^{\beta^- + N_{lm}^- - 1} \right) \end{aligned}$$

Due to the conjugacy, $\boldsymbol{\eta}$ can be marginalized:

$$\begin{aligned} P(\boldsymbol{A}, \boldsymbol{z}|\alpha, \beta^+, \beta^-) &= \int_0^1 P(\boldsymbol{A}, \boldsymbol{z}, \boldsymbol{\eta}|\alpha, \beta^+, \beta^-)\, d\boldsymbol{\eta} \\ &= P(\boldsymbol{z}|\alpha) \times \prod_{l \leq m} \left(\frac{\Gamma(\beta^+ + \beta^-)}{\Gamma(\beta^+)\Gamma(\beta^-)} \int_0^1 \eta_{lm}^{\beta^+ N_{lm}^+ - 1}(1 - \eta_{lm})^{\beta^- + N_{lm}^- - 1}\, d\eta_{lm} \right) \end{aligned}$$

With the beta-function defined as:

$$B(a, b) = \frac{\Gamma(a)\Gamma(b)}{\Gamma(a + b)} = \int_0^1 \theta^{a-1}(1 - \theta)^{b-1}\, d\theta,$$

we obtain:

$$P(\boldsymbol{A}, \boldsymbol{z}|\alpha, \beta^+, \beta^-) \;=\; P(\boldsymbol{z}|\alpha) \times \prod_{l \leq m} \frac{B(N^+_{lm} + \beta^+, N^-_{lm} + \beta^-)}{B(\beta^+, \beta^-)} \tag{G.3}$$

G.2 Poisson likelihood and Gamma prior

For a network where links represent discrete integer counts, the link probabilities can intuitively be modelled using a Poisson distribution. With the Gamma distribution acting as conjugate prior, the generative model becomes:

Links	$A_{ij} \sim Poisson(\eta_{z_i,z_j})$
Interactions	$\eta_{lm} \sim Gamma(a, b)$
Groups (infinite)	$\boldsymbol{z} \sim CRP(\alpha)$
Groups (finite)	$\boldsymbol{z} \sim Dirichlet\text{-}Categorical(\alpha)$

For the interaction between two groups l and m, the Gamma distribution gives:

$$Gamma(\eta_{lm}|a, b) = \frac{b^a \eta_{lm}^{a-1} e^{(-b\eta_{lm})}}{\Gamma(a)}$$

For all pair of groups this gives:

$$P(\boldsymbol{\eta}|a, b) = \prod_{l \leq m} \frac{b^a \eta_{lm}^{a-1} e^{(-b\eta_{lm})}}{\Gamma(a)} \tag{G.4}$$

For the link between two nodes i and j, the Poisson distribution gives:

$$Poisson(A_{ij}|\eta_{z_i z_j}) = e^{(-\eta_{z_i z_j})} \frac{\eta_{z_i z_j}^{A_{ij}}}{A_{ij}!}$$

Here A_{ij} is an integer count. Let π_{lm} be the sum of links between l and m while N_{lm} is the total pair of nodes between l and m. The distribution for links between all nodes now becomes:

$$P(\boldsymbol{A}|\boldsymbol{\eta}) = \frac{1}{\prod_{i \leq j}(A_{ij}!)} \prod_{l \leq m} \left(e^{(-\eta_{lm} N_{lm})} \eta_{lm}^{\pi_{lm}} \right) \tag{G.5}$$

By joining G.4 and G.5 we obtain the following joint distribution:

$$P(\boldsymbol{A}, \boldsymbol{z}, \boldsymbol{\eta}|\alpha, a, b) \;=\; P(\boldsymbol{z}|\alpha) \times P(\boldsymbol{A}|\boldsymbol{\eta}) \times P(\boldsymbol{\eta}|a, b)$$

$$= P(\boldsymbol{z}|\alpha) \times \frac{1}{\prod_{i\leq j}(A_{ij}!)} \prod_{l\leq m} \left(e^{(-\eta_{lm} N_{lm})} \eta_{lm}^{\pi_{lm}} \right) \prod_{l\leq m} \frac{b^a \eta_{lm}^{a-1} e^{(-b\eta_{lm})}}{\Gamma(a)}$$

$$= P(\boldsymbol{z}|\alpha) \times \frac{1}{\prod_{i\leq j}(A_{ij}!)} \prod_{l\leq m} \left(e^{-\eta_{lm}(N_{lm}+b)} \eta_{lm}^{\pi_{lm}+a-1} \frac{b^a}{\Gamma(a)} \right)$$

By integrating over $\boldsymbol{\eta}$ we obtain:

$$P(\boldsymbol{A}, \boldsymbol{z}|\alpha, a, b) = \int_0^\infty P(\boldsymbol{A}, \boldsymbol{z}, \boldsymbol{\eta}|\alpha, a, b)\, d\boldsymbol{\eta}$$

$$= P(\boldsymbol{z}|\alpha) \times \frac{1}{\prod_{i\leq j}(A_{ij}!)} \prod_{l\leq m} \left(\frac{b^a}{\Gamma(a)} \int_0^\infty e^{-\eta_{lm}(N_{lm}+b)} \eta_{lm}^{\pi_{lm}+a-1}\, d\eta_{lm} \right)$$

Given that

$$\int_0^\infty x^\beta e^{-\alpha x} dx = \frac{\Gamma(\beta+1)}{\alpha^{\beta+1}},$$

we obtain:

$$P(\boldsymbol{A}, \boldsymbol{z}|\alpha, a, b) = P(\boldsymbol{z}|\alpha) \times \frac{1}{\prod_{i\leq j}(A_{ij}!)} \prod_{l\leq m} \left(\frac{b^a}{\Gamma(a)} \frac{\Gamma(\pi_{lm}+a)}{(N_{lm}+b)^{\pi_{lm}+a}} \right) \quad \text{(G.6)}$$

G.3 Categorical likelihood and Dirichlet prior

For a network where links represent discrete categories, the link probabilities can intuitively be modelled using a Categorical distribution. With the Dirichlet distribution acting as conjugate prior, the generative model becomes:

Links	$A_{ij} \sim Categorical(\eta_{z_i,z_j})$
Interactions	$\eta_{lm} \sim Dirchlet(\beta)$
Groups (infinite)	$\boldsymbol{z} \sim CRP(\alpha)$
Groups (finite)	$\boldsymbol{z} \sim Dirichlet\text{-}Categorical(\alpha)$

For the interaction between two groups l and m, the Dirichlet distribution gives:

$$Dirichlet(\eta_{lm}|\beta) = \frac{\Gamma(\sum_{k=1}^K(\beta_k))}{\prod_{k=1}^K(\Gamma(\beta_k))} \prod_k^K \eta_{lm,k}^{\beta_k-1}$$

For all pair of groups this gives:

$$P(\boldsymbol{\eta}|\beta) = \prod_{l\leq m} \left(\frac{\Gamma(\sum_{k=1}^K(\beta_k))}{\prod_{k=1}^K(\Gamma(\beta_k))} \prod_k^K \eta_{lm,k}^{\beta_k-1} \right) \quad \text{(G.7)}$$

For the link between two nodes i and j, the categorical distribution over all K discrete categories yields:

$$Categorical(A_{ij}|\eta_{z_i z_j}) = \prod_{k=1}^{K} \eta_{z_i z_j,k}^{\delta(A_{ij,k}=1)},$$

where $\delta(A_{ij,k} = 1)$ evaluates to 1 if a link of category k exists between node i and j (with link-probability represented by $\eta_{z_i z_j,k}$). For all links we get:

$$P(\boldsymbol{A}|\boldsymbol{\eta}) = \prod_{l \leq m} \prod_{k=1}^{K} \eta_{z_i z_j,k}^{N_{lm}^k} \tag{G.8}$$

where N_{lm}^k is the number of links between group l and m belonging to category k.

By joining G.7 and G.8 we obtain the following joint distribution:

$$\begin{aligned} P(\boldsymbol{A}, \boldsymbol{z}, \boldsymbol{\eta}|\alpha, \beta) &= P(\boldsymbol{z}|\alpha) \times P(\boldsymbol{A}|\boldsymbol{\eta}) \times P(\boldsymbol{\eta}|\beta) \\ &= P(\boldsymbol{z}|\alpha) \times \prod_{l \leq m} \left(\frac{\Gamma(\sum_{k=1}^{K}(\beta_k))}{\prod_{k=1}^{K}(\Gamma(\beta_k))} \prod_{k=1}^{K} \eta_{lm,k}^{\beta_k + N_{lm}^k} - 1 \right) \end{aligned}$$

Due to the conjugacy, $\boldsymbol{\eta}$ can be marginalized:

$$\begin{aligned} P(\boldsymbol{A}, \boldsymbol{z}|\alpha, \beta) &= \int P(\boldsymbol{A}, \boldsymbol{z}, \boldsymbol{\eta}|\alpha, \beta)\, d\boldsymbol{\eta} \\ &= P(\boldsymbol{z}|\alpha) \times \prod_{l \leq m} \left(\frac{\Gamma(\sum_{k=1}^{K}(\beta_k))}{\prod_{k=1}^{K}(\Gamma(\beta_k))} \int \prod_{k=1}^{K} \eta_{lm,k}^{\beta_k + N_{lm}^k} - 1\, d\eta_{lm} \right) \end{aligned}$$

For the Dirichlet distribution it holds that:

$$\int Dirichlet(\theta|\beta)\, d\theta = 1 \Leftrightarrow \int \prod_{k=1}^{K} \theta^{\beta_k - 1}\, d\theta = \frac{\prod_{k=1}^{K}(\Gamma(\beta_k))}{\Gamma(\sum_{k=1}^{K} \beta_k)}$$

by which we obtain:

$$P(\boldsymbol{A}, \boldsymbol{z}|\alpha, \beta) = P(\boldsymbol{z}|\alpha) \times \prod_{l \leq m} \frac{\Gamma(\sum_{k=1}^{K} \beta_k)}{\prod_{k=1}^{K}(\Gamma(\beta_k))} \frac{\prod_{k=1}^{K}(\Gamma(\beta_k + N_{lm}^k))}{\Gamma(\sum_{k=1}^{K}(\beta_k + N_{lm}^k))} \tag{G.9}$$

G.4 Normal likelihood and Normal-Inverse-Gamma prior

For a network where weights are real values, the link probabilities can intuitively be modelled using a Normal distribution. The Normal-Inverse-Gamma distribution can act as conjugate prior to a normal distribution with unknown mean and variance. The generative model hence becomes:

Links	$A_{ij} \sim Normal(M_{z_i z_j}, \sigma^2_{z_i z_j})$
Interactions	$M_{lm}, \sigma^2_{lm} \sim Normal\text{-}Inverse\text{-}Gamma(\mu, \lambda, a, b)$
Groups (infinite)	$\boldsymbol{z} \sim CRP(\alpha)$
Groups (finite)	$\boldsymbol{z} \sim Dirichlet\text{-}Categorical(\alpha)$

Link probability between any pair of nodes i and j is given as:

$$P(A_{ij}|M_{z_i z_j}, \sigma^2_{z_i z_j}) = \frac{1}{\sigma_{z_i z_j}\sqrt{2\pi}} exp\left[-\frac{(A_{ij} - M_{z_i z_j})^2}{2\sigma^2_{z_i z_j}}\right]$$

Between group l and m: Let N_{lm} be the total number of links, let N^{sum}_{lm} be the sum of the weights of those links, let $N^{\overset{sum}{sqrt}}_{lm}$ be the sum of the squared weights of the links, and let $^{node}_{pairs}$ be the total number of pairs of nodes. We can now write the distribution for all pair of nodes as follows:

$$\begin{aligned}
P(\boldsymbol{A}|M, \sigma^2) &= \prod_{i \leq m} \left(\frac{1}{\sigma_{z_i z_j}\sqrt{2\pi}} exp\left[-\frac{(A_{ij} - M_{z_i z_j})^2}{2\sigma^2_{z_i z_j}}\right]\right) \\
&= \prod_{l \leq m} \left(\left(\frac{1}{\sqrt{2\pi}}\right)^{N_{lm}} \left(\frac{1}{\sigma_{lm}}\right)^{N_{lm}} exp\left[-\frac{N^{\overset{sum}{sqrt}}_{lm} + M^2_{lm} N_{lm} - 2N^{sum}_{lm} M_{lm}}{2\sigma^2_{lm}}\right]\right) \\
&= \left(\frac{1}{\sqrt{2\pi}}\right)^{^{node}_{pairs}} \prod_{l \leq m} \left(\left(\frac{1}{\sigma_{lm}}\right)^{N_{lm}} exp\left[-\frac{N^{\overset{sum}{sqrt}}_{lm} + M^2_{lm} N_{lm} - 2N^{sum}_{lm} M_{lm}}{2\sigma^2_{lm}}\right]\right)
\end{aligned} \tag{G.10}$$

The prior must depend on both M_{lm} and σ^2_{lm}. If these are not independent, the prior becomes:

$$P(M_{lm}, \sigma^2_{lm}) = P(M_{lm}|\sigma^2_{lm}) P(\sigma^2_{lm})$$

A conjugate prior for the normal distribution with unknown mean and variance is a Normal-Inverse-Gamma distribution, such that:

$$\begin{aligned}
P(\sigma^2_{lm}) &= Inverse\text{-}Gamma(\sigma^2_{lm}|a, b) \\
P(M_{lm}|\sigma^2_{lm}) &= Normal(\mu, \sigma^2_{lm}/\lambda)
\end{aligned}$$

stating that the dependence is a linear relation between the variances of the normal distributions for $M_{z_i z_j}$ and A_{ij}.

The Normal-Inverse-Gamma prior becomes:

$$\begin{aligned} P(M_{lm}, \sigma^2_{lm} | \mu, \lambda, a, b) &= \frac{\sqrt{\lambda}}{\sigma_{lm}\sqrt{2\pi}} \frac{b^a}{\Gamma(a)} \left(\frac{1}{\sigma^2_{lm}}\right)^{a+1} exp\left[-\frac{2b + \lambda(M_{lm} - \mu)^2}{2\sigma^2_{lm}}\right] \\ &= \frac{\sqrt{\lambda}}{\sqrt{2\pi}} \frac{b^a}{\Gamma(a)} \left(\frac{1}{\sigma^2_{lm}}\right)^{a+1\frac{1}{2}} exp\left[\frac{-2b - \lambda(M_{lm} - \mu)^2}{2\sigma^2_{lm}}\right] \end{aligned}$$

Letting $\overset{cluster}{pairs}$ denote the total pair of clusters, the prior over all pairs of clusters is:

$$P(M, \sigma^2 | \mu, \lambda, a, b) =$$

$$\left(\frac{\sqrt{\lambda}}{\sqrt{2\pi}} \frac{b^a}{\Gamma(a)}\right)^{\overset{cluster}{pairs}} \prod_{l \leq m} \left(\left(\frac{1}{\sigma^2_{lm}}\right)^{a+1\frac{1}{2}} exp\left[-\frac{2b + \lambda(M_{lm} - \mu)^2}{2\sigma^2_{lm}}\right]\right) \quad \text{(G.11)}$$

Let C denote the product of the constants in the likelihood and prior:

$$C = \left(\frac{1}{\sqrt{2\pi}}\right)^{\overset{node}{pairs}} \left(\frac{\sqrt{\lambda}}{\sqrt{2\pi}} \frac{b^a}{\Gamma(a)}\right)^{\overset{cluster}{pairs}}$$

Joining the prior and likelihood yields:

$$P(\boldsymbol{A}, \boldsymbol{z}, M, \sigma^2 | \alpha, \mu, \lambda, a, b) = P(\boldsymbol{z}|\alpha) \times P(\boldsymbol{A}|M, \sigma^2) \times P(M, \sigma^2 | \mu, \lambda, a, b)$$

$$\begin{aligned} &= P(\boldsymbol{z}|\alpha) \times C \times \prod_{l \leq m} \left(\left(\frac{1}{\sigma^2_{lm}}\right)^{\frac{1}{2}N_{lm} + a + 1\frac{1}{2}}\right. \\ &\left. exp\left[\frac{-N^{\overset{sum}{sqrt}}_{lm} - M^2_{lm} N_{lm} + 2N^{sum}_{lm} M_{lm} - 2b - \lambda M^2_{lm} - \lambda\mu^2 + 2\mu M_{lm}}{2\sigma^2_{lm}}\right]\right) \end{aligned} \quad \text{(G.12)}$$

From this expression we can in turn marginalize M and σ^2.

Only the exponent in G.12 depends on M. To marginalize M we hence integrating over the exponent for all M_{lm} for all clusters $l \leq m$.

Consider the integral for a normal distribution:

$$\int \frac{1}{\sqrt{2\pi\hat{\sigma}^2}} exp\left[-\frac{(\hat{x}-\hat{\mu})^2}{2\hat{\sigma}^2}\right] dx = 1$$

$$\Leftrightarrow \quad \int \frac{1}{\sqrt{2\pi\hat{\sigma}^2}} exp\left[-\frac{\hat{x}^2}{2\hat{\sigma}^2} + \frac{\hat{x}\hat{\mu}}{\hat{\sigma}^2} - \frac{\hat{\mu}^2}{2\hat{\sigma}^2}\right] dx = 1 \tag{G.13}$$

$$\Leftrightarrow \quad \int exp\left[-\frac{\hat{x}^2}{2\hat{\sigma}^2} + \frac{\hat{x}\hat{\mu}}{\hat{\sigma}^2}\right] dx = \sqrt{2\pi\hat{\sigma}^2}\, exp\left[\frac{\hat{\mu}^2}{2\hat{\sigma}^2}\right] \tag{G.14}$$

By defining

$$\hat{\alpha} = \frac{1}{2\hat{\sigma}^2} \quad , \quad \hat{\beta} = \frac{\hat{\mu}}{\hat{\sigma}^2} \quad , \quad \hat{\gamma} = \frac{\hat{\mu}^2}{2\hat{\sigma}^2} \tag{G.15}$$

The exponent in G.13 can be written as:

$$exp\left[-\hat{\alpha}\hat{x}^2 + \hat{\beta}\hat{x} - \hat{\gamma}\right]. \tag{G.16}$$

Restructuring the exponent in expression G.12 we can define α, β and γ;

$$exp\left[-\overbrace{\frac{(N_{lm}+\lambda)}{2\sigma_{lm}^2}}^{\alpha} M_{lm}^2 + \overbrace{\frac{(N_{lm}^{sum}+\mu)}{\sigma_{lm}^2}}^{\beta} M_{lm} - \overbrace{\frac{N_{lm}^{sum_{sqrt}} + 2b + \lambda\mu^2}{2\sigma_{lm}^2}}^{\gamma}\right] \quad ,$$

such that the exponent can be written as

$$exp\left[-\alpha M_{lm}^2 + \beta M_{lm} - \gamma\right] \tag{G.17}$$

which can be used to map between the exponents G.17 and G.16, such that:

$$M_{lm} = \hat{x} \quad , \quad \alpha = \hat{\alpha} \quad , \quad \beta = \hat{\beta},$$

From the expressions G.15 we get:

$$\hat{\sigma}^2 = \frac{1}{2\hat{\alpha}} \quad , \quad \hat{\mu} = \frac{\hat{\beta}}{2\hat{\alpha}}$$

which can be used in order to express $\hat{\sigma}^2$ and $\hat{\mu}$ by σ^2 and μ:

$$\hat{\sigma}^2 = \frac{1}{2\alpha} = \frac{\sigma_{lm}^2}{N_{lm}+\lambda} \quad , \quad \hat{\mu} = \frac{\beta}{2\alpha} = \frac{N_{lm}^{sum}+\mu}{N_{lm}+\lambda} \tag{G.18}$$

Integrating over M in G.17 as in G.14 gives:

$$\begin{aligned}\int exp\left(-\alpha M_{lm}^2 + \beta M_{lm} - \gamma\right) dM \quad &= \quad \sqrt{2\pi\hat{\sigma}^2}\; exp\left(\frac{\hat{\mu}^2}{2\hat{\sigma}^2}\right)\; exp\left(-\gamma\right) \\ = \sqrt{\pi\frac{2\sigma_{lm}^2}{N_{lm}+\lambda}}\; exp\left[\frac{(N_{lm}^{sum}+\mu)^2}{(N_{lm}+\lambda)2\sigma_{lm}^2}\right]\; exp\left[-\frac{N_{lm}^{sum_{sqrt}}+2b+\lambda\mu^2}{2\sigma_{lm}^2}\right]\end{aligned} \tag{G.19}$$

The joint distribution given by equation G.12 can now be expressed without M_{lm}:

$$\begin{aligned}P(\boldsymbol{A}, \boldsymbol{z}, \sigma^2|\alpha,\mu,\lambda,a,b) \quad &= \quad P(\boldsymbol{z}|\alpha) \times C \times \prod_{l\leq m}\left(\left(\frac{1}{\sigma_{lm}^2}\right)^{\frac{1}{2}N_{lm}+a+1}\right. \\ \left.\times\sqrt{\frac{2\pi}{N_{lm}+\lambda}} \times exp\left[\frac{1}{\sigma_{lm}^2}\frac{1}{2}\left(\frac{(N_{lm}^{sum}+\mu)^2}{N_{lm}+\lambda} - \left(N_{lm}^{sum_{sqrt}}+2b+\lambda\mu^2\right)\right)\right]\right)\end{aligned} \tag{G.20}$$

The integral of the inverse gamma distribution is given by:

$$\begin{aligned}\int \frac{\hat{\beta}^{\hat{\alpha}}}{\Gamma(\hat{\alpha})} x^{-\hat{\alpha}-1}\; exp(-\frac{\hat{\beta}}{x})\; dx \;&=\; 1 \\ \Leftrightarrow \qquad \int x^{-\hat{\alpha}-1} exp(-\frac{\hat{\beta}}{x})\; dx \;&=\; \frac{\Gamma(\hat{\alpha})}{\hat{\beta}^{\hat{\alpha}}}\end{aligned} \tag{G.21}$$

From equation G.22 integrate over σ^2 by setting:

$$\begin{aligned}\hat{\alpha} \;&=\; -\frac{1}{2}N_{lm} - a \\ \hat{\beta} \;&=\; \frac{1}{2}\left(\frac{(N_{lm}^{sum}+\mu)^2}{N_{lm}+\lambda} - \left(N_{lm}^{sum_{sqrt}}+2b+\lambda\mu^2\right)\right)\end{aligned}$$

Inserted into the form in G.21 we get the final expression for the posterior, without M or σ^2, where α denotes the concentration parameter of the clustering prior:

$$P(\boldsymbol{A}, \boldsymbol{z}|\alpha, \mu, \lambda, a, b) \quad = \quad P(\boldsymbol{z}|\alpha) \times C \times \prod_{l \leq m} \left(\times \sqrt{\frac{2\pi}{N_{lm} + \lambda}} \times \frac{\Gamma\left(-\frac{1}{2} N_{lm} - a\right)}{\left(\frac{1}{2} \frac{(N_{lm}^{sum} + \mu)^2}{N_{lm} + \lambda} - (N_{lm}^{sqrt{sum}} + 2b + \lambda \mu^2)\right)^{\left(-\frac{1}{2} N_{lm} - a\right)}} \right). \tag{G.22}$$

Appendix H

Supplementary information

H.1 Predictive Evaluation of Human Value Segmentations

Predictive Evaluation of Human Value Segmentations. Kristoffer Jon Albers, Morten Mørup, Mikkel N. Schmidt, and Fumiko Kano Glückstad (2017, Under review).

under review
©2017

Predictive evaluation of human value segmentations

Kristoffer Jon Albers[1], Morten Mørup[1], Mikkel N. Schmidt[1], Fumiko Kano Glüuckstad[2]

Abstract

While data-driven segmentation plays an important role in understanding and analysing patterns of associations within social survey data, comparing the quality of segmentations obtained by different methods remains challenging. In this paper we propose to quantify the quality of segmentations of human values using a proposed statistical framework, where the model fit of different segmentation methods are evaluated based on their capabilities to predict unmodelled hold-out data. By comparing clusterings of human values survey data from the forth round of European Social Study (ESS-4), we show that demographic markers such as age or country predicts better than random, yet are outperformed by data-driven segmentation methods. We present that a Bayesian version of Latent Class Analysis (LCA) outperforms the standard maximum likelihood LCA in predictive performance and is more robust for different number of clusters.

Keywords

Latent class analysis, statistical modeling, Prediction, human values.

Introduction

The recent trend of globalization facilitated by world-wide communication- and internet technologies has affected people's identity- and value formation, since common values can now be easily shared across geographical boundaries. This implies that cultural values within common regions and nations become differentiated if traditional local values are being mixed with universal values promoted by the fast paced globalization. Investigating the heterogeneities between and within nations has become a prominent

[1]Technical University of Denmark, Department of Applied Mathematics and Compute Science.
[2]Copenhagen Business School, Department of International Business Communication.

topic among researchers in various disciplines such as cross-cultural psychology, sociology, and marketing sciences.

The emerging heterogeneities are often investigated by utilizing various clustering methods — automated search procedures for partitioning a data set into groups of similar data points. In practice, clustering is often based on heuristic methods (Fraley and Raftery 2002), where the data is partitioned in order to maximize the between-cluster differences and/or minimize the within-cluster differences according to a given cost function. A popular example of this class of algorithms are centroid based clustering, such as k-means, in which data points are iteratively reallocated to clusters until no further improvement can be obtained. Another example is hierarchical agglomerative clustering, in which pairs of clusters are iteratively merged in order to optimize the chosen criterion, often being the shortest or average distance between clusters or the within-cluster variance (Ward 1963). Heuristic methods can be both conceptually intuitive and simple to apply, and often have very reasonable computational times. These methods, however, lack a statistical foundation, which limits the way relevant questions, such as determining an appropriate number of clusters, can be theoretically evaluated (Picard 2007).

Probabilistic based clustering methods exists, including mixture models such as Latent Class Analysis (LCA) (McCutcheon 1987), which has become one of the most widely used tools for conducting clustering analysis within many different research fields (Berzofsky et al. 2014). In the social sciences, LCA has been used to extract different patterns of people's behaviors, attitudes or value priorities (Szakolczai and Füstös 1998; Magun and Rudnev 2008, 2015; Moors and Vermunt 2007). In sociology and cross-cultural psychology, LCA has been applied to analyze patterns of various survey responses such as the European Social Survey and the World Value Survey (Eid et al. 2003; Magun et al. 2015; Kankaraš et al. 2010; Finch and Bronk 2011; Rudnev et al. 2014). Magun et al. (2015) applied an extended version of the traditional LCA, the so-called Factor Mixture Model (Muthén 2008) to analyze the response patterns of 21 question items in the Portrait Value Questionnaire (PVQ21) (Schwartz et al. 2001) available from the 4th round of the European Social Survey 2008-2009 (ESS) for 29 European countries (Jowell et al. 2007), and identified five clusters based on data from approximately 55 000 respondents.

In the aforementioned works, the number of clusters is most often identified based on a model selection criterion such as the Bayesian Information Criterion (BIC) or Akaike Information Criterion (AIC). Since a more complex model with more latent classes can always fit the data better, these criteria make a trade-off between the model fit and the complexity of the model in terms of the number of parameters, to avoid overfitting. While these criteria are valid asymptotically under certain assumptions, it is debatable which criterion to use. Nylund et al. (2007) compares several approaches to estimating the number of clusters in LCA and similar mixture models, and concludes that the (very computationally demanding) bootstrap likelihood ratio test (BLRT) performs best, followed by BIC; however, it is clear that the model selection criterion is important and has a strong influence on the estimated number of clusters.

One of the purposes in value segmentation is to identify subgroups of individuals who share behaviours or attitudes in a manner, such that they can be characterized as

accurately as possible by their latent class membership. From this point of view, it is important to estimate an appropriately large number of clusters of which members share homogeneous response patterns that are clearly distinguished from patterns indicated by other clusters. On the other hand, the *"assumption of within-segment homogeneity may be overly restrictive and result in a loss of explanatory and predictive power"* (Allenby et al. 1998). In other words, it is difficult to identify an appropriate number of clusters that are both specific and homogeneous, yet not so restrictive that their explanatory power is lost.

In an exploratory latent class analysis, the objective is often to identify a single clustering that best complies with the data, in order to provide a directly interpretable characterization of the response patterns in the data. In the classical maximum likelihood (ML) approach to LCA, the final result is the clustering which maximizes the probability of the data under the model. An issue with the ML approach is that it does not directly take into account the statistical uncertainty associated with the solution. If the number of clusters is low, such that each cluster has a substantial number of data associated with it, the clusters will be statistically well defined, whereas if the number of clusters is high relative to the number of data, the uncertainty may be substantial. An advantage of the Bayesian approach to data modeling, in contrast with ML, is that the final result is a posterior probability distribution of latent classes rather than a single clustering. If needed, the Bayesian posterior can still be summarized by a single clustering that is Bayes-optimal according to a specified utility function (Rastelli and Friel 2016); however, in terms of explanatory power, utilizing the uncertainty by averaging over the posterior distribution often yields better predictions.

In this paper we compare the classical maximum likelihood LCA (Lanza et al. 2007) with a Bayesian LCA in which the posterior uncertainty of the latent classes is taken into account. To simplify the presentation, we limit the discussion to binary data, but we note that extensions to categorical, ordered categorical, and nominal data is possible. We apply the ML and Bayesian LCA models to a human value questionnaire data set similar to the data analyzed by Magun et al. (2015). We show that the maximum likelihood and Bayesian approaches lead to similar clusterings, but that the Bayesian approach has superior predictive performance on held-out data because it incorporates uncertainty in its estimate. We further demonstrate how the Bayesian LCA can be used in a predictive approach to model order selection, in which an appropriate number of clusters is identified by optimizing the predictive performance on held-out data.

Data and method

Values play a central role for explaining individuals' belongings to social groups and the motivational basis of attitudes and behavior. Values have been studied by researchers in sociology, psychology and anthropology for portraying societies, organizations, and individuals.

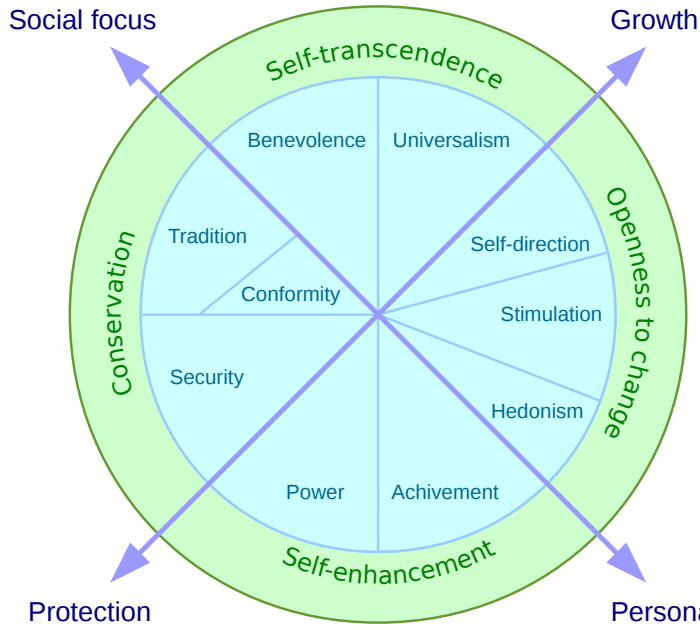


Figure 1. Circular model illustrating the relations between the 10 basic values of Schwartz' theory.

The schwartz dataset

Schwartz's theory of 10 basic values is one of the most widely applied value theories, and has been integrated among others in the World Value Survey (WVS) and the European Social Survey (ESS). According to Schwartz (2012), his theory is capable of capturing characteristics of value priorities both at an individual and at a societal level, and it also accommodates primary characteristics of values previously defined by prior theorists (e.g. Allport 1961; Feather 1995; Kluckhohn 1951; Morris 1956; Rokeach 1973).

The 10 basic values (Schwartz 2012; Smith and Schwartz 1997) are listed in the following: *self-direction*, *stimulation*, *hedonism*, *achievement*, *power*, *security*, *conformity*, *tradition*, *benevolence* and *universalism*. In Schwartz theory, these 10 basic values are organized in a circular model indicating two aspects of value relations: conflict values vs. congruent values (see Figure 1). For example, a person with a higher stimulation value who seeks an exciting and varied life may likely undermine the tradition- or security values (conflict relation). On the other hand, a person who prioritizes the achievement value in his/her life may prioritize the power value too (congruent relation). In Figure 1, the stimulation value and the tradition- or security values are located in opposing positions, whereas the achievement- and power values are next to each other. In other words, the circular model effectively surveys the conflict vs. congruent relations among these 10 basic values (Schwartz 2007, 2012).

Schwartz et al. (2001) further categorizes the 10 basic values into four superordinate values as follows: *openness to change*, *self-enhancement*, *conservation* and *self-transcendence* as shown in Figure 1. Among these, the values belonging to the category *openness to change* are opposed to the values belonging to the category *conservation*. In the same way, the values in the *self-enhancement* category are opposed to the values in the *self-transcendence* category.

This theory of 10 basic values has been assessed by a number of scientists (Bilsky et al. 2011; Davidov et al. 2011) who elaborate on Schwartz's theory to a dynamic model. The dynamic underpinning of the value structure describes that the values belonging to *self-enhancement* and *openness to change* directly regulate "how one expresses personal interests & characteristics" (Schwartz 2012), whereas the values belonging to *conservation* and *self-transcendence* regulate "how one relates socially to others and affects them" (Schwartz 2012). In addition, the values belonging to *self-enhancement* and *conservation* are anxiety-based values that prevent loss of goals (a self-protection against threats), whereas the values belonging to *openness to change* and *self-transcendence* are anxiety-free values that promote gain of goals (self-expansion and growth) (Schwartz 2012).

Considering these aspects of Schwartz's theory, it is expected that the explorative pattern analysis of these value priorities indicated by individuals uncover heterogeneous structures of societies that are more or less invisible for the traditional cross-cultural comparative analysis. Our quantitative data analysis employs the fourth round of the European Social Study, ESS-4: 2008-2009 (Jowell et al. 2007) accessible from the ESS organization*, in order to be able to contrast our results with existing works such as Magun et al. (2015). The dataset contains responses from approximately 55 000 respondents from 29 European countries. The questionnaire includes a simplified version of the Portrait Values Questionnaire (PVQ) developed by Schwartz et al. (2001) that consists of 21 question items portraying people expressing different goals, aspirations, or wishes that point implicitly to the importance of a value (Schwartz 2012).

Specifically, 21 questions are classified into 10 basic values as follows: *Self-Direction* (Important to think new ideas and being creative, Important to make own decisions and be free); *Stimulation* (Important to try new and different things in life, Important to seek adventures and have an exciting life); *Hedonism* (Important to have a good time, Important to seek fun and things that give pleasure); *Security* (Important to live in secure and safe surroundings; Important that government is strong and ensures safety); *Conservation* (Important to do what is told and follow rules; Important to behave properly); *Tradition* (Important to follow traditions and customs; Important to be humble and modest, not draw attention); *Benevolence* (Important to help people and care for others well-being; Important to be loyal to friends and devote to people close); *Universalism* (Important to understand different people; Important to care for nature and environment; Important that people are treated equally and have equal opportunities); *Achievement* (Important to show abilities and be admired; Important to be successful

*http://www.europeansocialsurvey.org/

and that people recognize achievements); *Power* (Important to be rich, have money and expensive things, Important to get respect from others).

From the original dataset accessible from ESS-4, respondents missing a response to any of these 21 question items were removed, which resulted in 51 641 respondents used in our analysis. We randomly select 80 percent of the respondents as training data with the remaining 20 percent used as hold out data for evaluating the predictive performance of the models. The answers to the 21 questions in the ESS-questionnaire are given by the following six ordered categories:

1, Very much like me. 2, Like me. 3, Somewhat like me. 4, A little like me.	}	1, Positive response
5, Not like me. 6, Not at all like me.	}	0, Negative response

Here categories 1 through 4 semantically represents positive responses while category 5 and 6 represents negative responses (see also Glückstad et al. 2016). As shown in Table 1, the asymmetry between positive and negative responses in the data is pronounced, with an average of 87 percent positive responses across all question items. As our primary objective is to highlight differences between the maximum likelihood and Bayesian LCA methods, we limit the discussion to the corresponding models for binary observations. We find it reasonable to binarize the data with the threshold for the two categories set to separate between positive and negative responses, as argued in Glückstad et al. (2016). This allows us to analyze the data using binary LCA methods, but we emphasize that similar results could be obtained by analyzing the ordered categorical data.

Latent class analysis

Within social sciences, observed data often exhibit some form of heterogeneity even though the underlying source cannot be observed directly. The goal of LCA is then to partition the population of data items into groups of similar items, based on the latent concepts that cause observed correlations within the data. For the ESS-data, the clustering problem becomes to split the N respondents into K clusters, based on the structure of their response patterns for the $Q = 21$ questions. The binarized ESS data can be considered as a binary matrix with with N rows (the respondents) and 21 columns (the question items):

$$\boldsymbol{X} = \begin{bmatrix} x_{1,1} & x_{1,2} & \dots & x_{1,21} \\ x_{2,1} & x_{2,2} & \dots & x_{2,21} \\ \vdots & \vdots & \ddots & \vdots \\ x_{N,1} & x_{N,2} & \dots & x_{N,21} \end{bmatrix}. \tag{1}$$

We examine and compare two LCA approaches for modeling the ESS-data:

Value group	Question	Mean of six-scale	Percent positive
Openness to change	Creative	2.59	90.54
	New things	2.96	84.12
	Good time	2.95	84.35
	Own decisions	2.21	95.79
	Adventures	3.88	**59.25**
	Fun	3.09	81.38
Conservatism	Secure	2.27	93.81
	Follow rules	2.99	82.57
	Modest	2.72	89.19
	Safety	2.27	94.13
	Behaviour	2.56	91.92
	Traditions	2.60	89.62
Self-transendence	Equality	2.09	96.74
	Understand	2.38	95.86
	Help people	2.20	**97.73**
	Friends	**1.97**	98.48
	Nature	2.13	97.42
Self-enhancement	Rich	**3.89**	59.57
	Abilities	3.02	82.37
	Success	3.03	82.64
	Respect	3.06	81.47
Mean of all responses		2.71	87.09

Table 1. The 21 questions of the ESS dataset, split into four value groups. The table lists the mean when considering the six level answer categories as scalable continuous data and lists the percentage of positive answers for each question item.

1. The classical maximum likelihood LCA (McCutcheon 1987) using the expectation-maximization (EM) algorithm, which has proven useful in many fields (Berzofsky et al. 2014). We utilize the SAS implementation of LCA presented in (Lanza et al. 2007).
2. A Bayesian version of LCA, particularly a finite Bayesian mixture model (BMM) with Bernoulli likelihood and Beta prior, using a Markov chain Monte Carlo procedure for inferring the posterior distribution.

In the LCA models, the data items are assumed being generated from a mixture of probability distributions, where each mixture component represents a latent cluster. In the case of binary observations, the likelihood can be expressed in terms of the latent

class memberships, z_n and the response probabilities $\eta_{k,q}$ as

$$p(\boldsymbol{X}|\eta, z) = \prod_{n=1}^{N} \prod_{q=1}^{Q} \text{Bernoulli}(x_{n,q}; \eta_{z_n,q}), \tag{2}$$

where N is the number of respondents, Q is the number of questions, and z_n is the latent cluster assignment of respondent n. Thus, each binary response, $x_{n,q}$ is modeled by a Bernoulli distribution (biased coin flip) with parameter $\eta_{z_n,q}$ specific for the particular question and latent cluster. Equivalently, the likelihood can be expressed in terms of the probability of membership of each latent class, γ_k,

$$p(\boldsymbol{X}|\eta, \gamma) = \prod_{n=1}^{N} \left[\sum_{k=1}^{K} \gamma_k \prod_{q=1}^{Q} \eta_{k,q}^{x_{n,q}} (1 - \eta_{k,q})^{1-x_{n,q}} \right]. \tag{3}$$

The maximum likelihood estimate of the LCA is usually computed using an EM procedure which iteratively optimizes the expected log-likelihood function.

Bayesian LCA

The Bayesian LCA differs from the classical LCA by introducing prior distributions on the parameters. Here, we choose vague (non-informative) flexible priors, which only influence the results minimally. For the response probability parameters η, we use a separate Beta distribution for each question. The Beta distribution has two so-called hyper-parameters, which we denote β_q^+ and β_q^-, that flexibly can specify a suitable distribution on $\eta_{k,q} \in [0, 1]$. To choose the hyper-parameters, we take a hierarchical Bayesian approach, and endow them with a vague hyper-prior $p(\beta_{k,q}) \propto 1/\beta_{k,q}$. This allows us to effectively let the data define appropriate prior distributions for the response probabilities for each question and for each cluster. An advantage of the Beta priors is that the Beta distribution is conjugate to the Bernoulli likelihood, which makes it possible to analytically marginalize (integrate out) the parameters, significantly simplifying the inference procedure. For the cluster assignments z we use a standard Dirichlet-Categorical prior, which has a single hyper-parameter, the so-called concentration parameter α that governs the cluster size distribution. Again, we take a hierarchical Bayesian approach, and endow α with a vague prior, $p(\alpha) \propto 1/\alpha$.

The Bayesian LCA model can be summarized by the following generative process:

$$z_n \sim \text{Dirichlet-Categorical}(\alpha) \qquad \text{Clustering of respondents} \tag{4}$$

$$\eta_{k,q} \sim \text{Beta}(\beta_q^+, \beta_q^-) \qquad \text{Response probability} \tag{5}$$

$$x_{n,q} \sim \text{Bernoulli}(\eta_{z_n q}) \qquad \text{Response} \tag{6}$$

The model parameters are inferred using Markov chain Monte Carlo, simulating samples from the posterior distribution $p(\boldsymbol{z}|\boldsymbol{X})$. The cluster labels $\boldsymbol{z}$ are inferred using Gibbs sampling, while the hyper-parameters $\alpha, \beta_1^+, ..., \beta_Q^+, \beta_1^-, ..., \beta_Q^-$ are individually inferred using a Metropolis-Hastings sampling procedure. Technical details regarding the model specification and inference procedure are described in the Appendix.

Evaluating predictive performance

A key advantage of using a generative probabilistic approach to clustering is that the model provides a principled approach to evaluating the model fit by prediction on held-out test data. Given a model fitted on data $\boldsymbol{X}$, the predictive likelihood of the 21 binary observations $\boldsymbol{x}^*$ from a new respondent is given by

$$p(\boldsymbol{x}^*|z, \boldsymbol{X}) = \sum_{k=1}^{K} \hat{\gamma}_k \prod_{q=1}^{Q} \hat{\eta}_{k,q}^{x_q^*} (1 - \hat{\eta}_{k,q})^{1-x_q^*}, \tag{7}$$

$$\hat{\eta}_{k,q} = \frac{n_{k,q} + \beta_q^+}{m_k + \beta_q^+ + \beta_q^-}, \quad \hat{\gamma}_k = \frac{m_k + \frac{\alpha}{K}}{N}, \tag{8}$$

where $n_{k,q}$ is the number of positive responses in cluster k on feature q, and m_k is the size of cluster k. As a measure of model fit, we average the logarithm of this expression over the held-out test observations, to yield an estimate of the predictive log-likelihood. The predictive log-likelihood can be used to estimate the appropriate number of clusters, by fitting models with a varying number of clusters and comparing their predictive power.

Results and Analysis

Segmentations of respondents can be obtained in multiple ways. First we explore the data by partitioning the respondents according to the demographics parameters within the dataset itself. This is illustrated using respondents age, country of origin and combinations thereof. We evaluate how well groups identified by this approach captures characteristics of shared value priorities as expected (Schwartz 2003).

Using the predictive framework, the demographics based segmentations are compared with the data-driven modelling techniques. We use the Predictive log-likelihood on hold-out data as the measure to compare the predictive performance of the different clustrering techniques. Segmentations obtained by standard maximum likelihood and Bayesian LCA are furthermore compared both in terms of their predictive performance and according to their capability of partitioning the respondents according to their response patterns, value priorities and demographics.

Segmentations based on demographics

Based on the respondents' answers to the 21 questions, they can individually be positioned on the two value dimensions spanned by the Schwartz circle, as illustrated in Figure 2 for all respondents in the training data. The position on the horizontal axis is computed as the sum of positive responses to the 6 questions associated with the *openness to change* value group subtracted by the sum of positive responses to the 6 questions associated with the *conservation* value group. The position on the vertical axis is likewise computed as the sum of positive responses to the 5 question associated with *self-transcendence* subtracted by the sum of positive responses to the 4 questions associated with *self-enhancement*. The figure is coloured to indicate the number of respondents that share the same position in both dimensions.

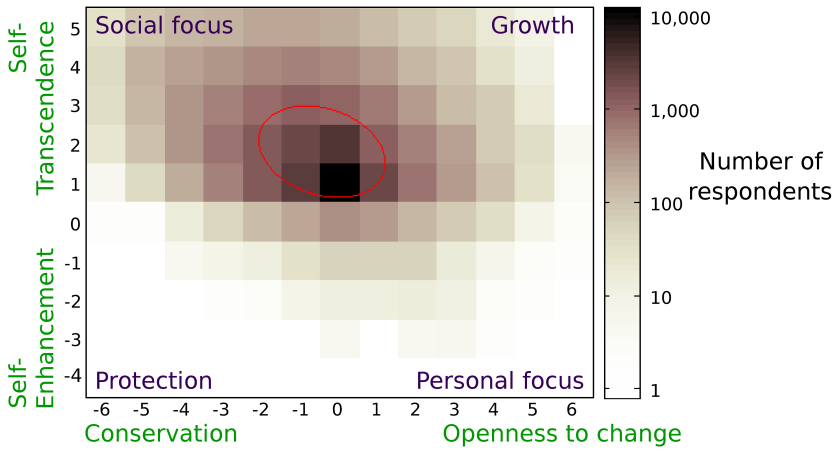


Figure 2. The entire training data projected onto the two value-dimensions of the Schwartz circle in Figure 1. Color intensity indicates number of respondents. The ellipse indicates one standard deviation from the mean of all respondents.

The most common shared position is (0,1), corresponding to an equal number of positive responses to the *conservation* and *openness to change* value groups and one more positive answer in the *self-transcendence* value group than in *self-enhancement*. Because the *self-transcendence* value group is associated with one more question item than *self-enhancement* respondents that answers positive to all 21 questions will be positioned here.

Schwartz (2003) expects positive correlation between age and conservation values, i.e., older people tend to have stronger conservation values and self-transcendence values and vice versa (see also Tyler and Schuller 1991; Veroff et al. 1984). This effect is clearly shown in Figure 3 where the respondents are partitioned according to age-groups. The position on the value dimensions are here computed as the average response for all respondents in a given age-group. In the figure, the age-groups are further subpartitioned according to four geographical regions; *Nordic countries*, *West European countries*, *Mediterranean countries* and *Post-Communist countries* as classified by Magun et al. (2015). The geographical subpartitioning illustrates that the diversity of value-orientation is similar within all regions: The young age-groups share values associated with personal focus while older age-groups share values associated with social focus.

Although the figure shows that there exists between-region similarities on the personal to social focus diagonal (that can be explained by age), it indicates that there is between-region diversity on the protection to growth diagonal. Figure 4 further supports that this between-region diversity can be explained by nationality. Here respondents are partitioned according to nationality only. The figure indicates that respondents from post-communist countries commonly share values associated with personal protection while respondents from west and north European countries share more growth oriented values.

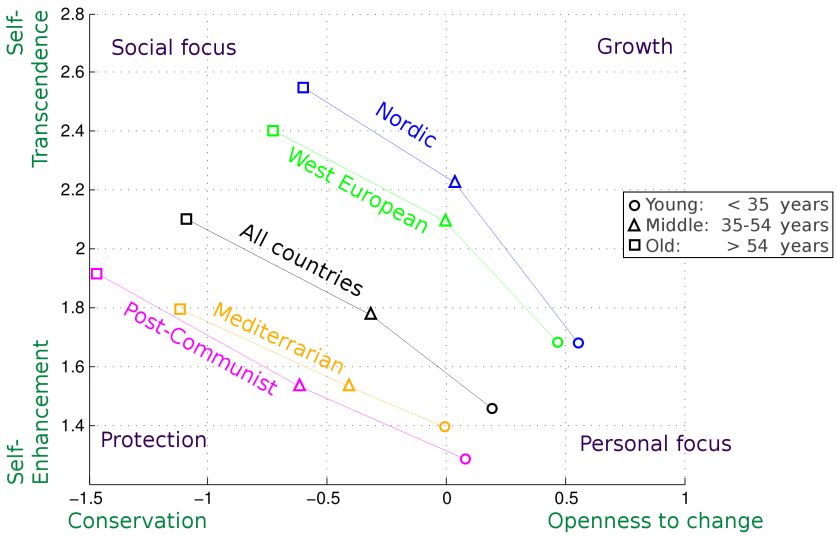


Figure 3. Clustered according to age groups (young, middle and old), the training data is projected on the value axis of the Schwartz circle shown in figure 1. Respondents in the data without an associated age are ignored.

From Figure 3 and Figure 4 the two demographic markers (age-group and geographical region) seem fairly capable of separating the data according to the value dimensions of Schwartz's theory.

Segmentations inferred by LCA

Both the traditional and the Bayesian LCA models were fitted to the training data for the following number of clusters $K = \{5, 10, 20, ..., 70\}$. As the result of the inference for the LCA models can be influcence by initial conditions, the models were fitted 5 times with different random initial conditions for each K, resulting in five independently inferred clusterings for each K.

To asses the stability of the solutions, Figure 5 compares the sizes of the inferred clusters, both when varying K and for two independent clusterings fitted for the same K. The figure indicates that there are slight differences in the size distribution for the same K. For $K = 5$ and $K = 10$ the distribution of cluster sizes seems to be rather similar between the two models. For higher K standard LCA seems to relatively assign more respondents to the largest cluster.

Evaluating the predictive performance

The predictive log-likelihood for various number of clusters is shown in Figure 6, where predictions were made on the 20 percent held-out data. The figure shows the average over five re-runs of both standard and Bayesian LCA for the different number of clusters.

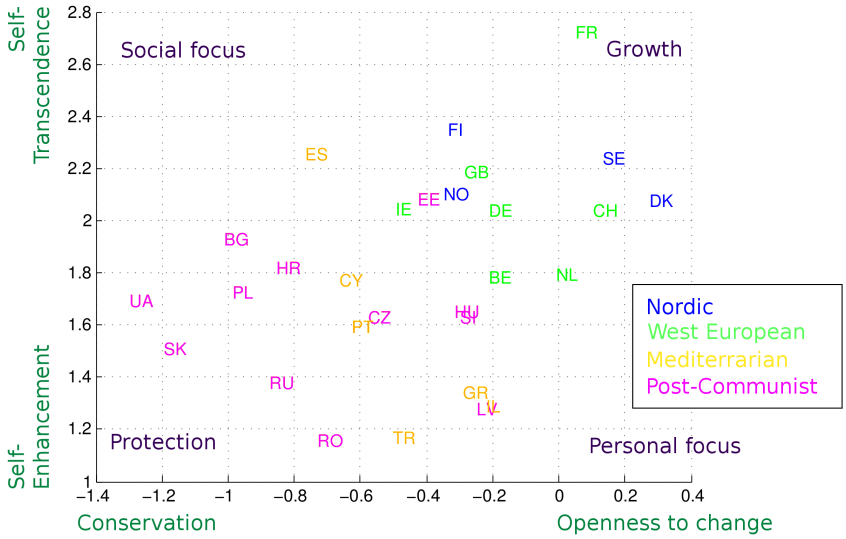


Figure 4. Clustered according to country, the training data is projected on the value axis of the Schwartz circle shown in Figure 1. Each country is positioned according to the average value for all respondents in the training data. Country names are abbreviated as in Figure 9.

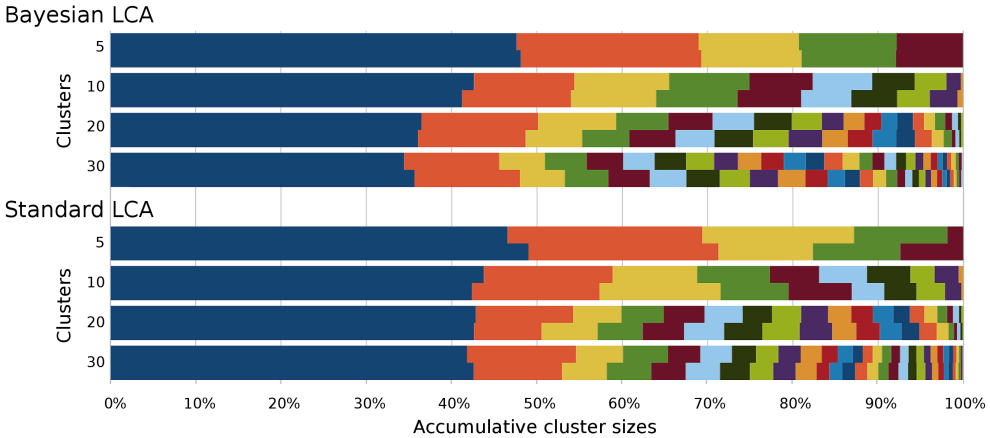


Figure 5. Stacked cluster sizes of clusterings inferred by multiple runs of the models on the training data. The figure shows results for two independent runs performed for $K = \{5, 10, 20, 30\}$ number of clusters respectively for the models.

For comparison, the predictive performances of clusterings obtained from K-means (using city-block distance measure) and Hierarchical clustering (using the Ward linkage function) are shown. As a baseline, the predictive performance using a random partition is included. Furthermore, the figure shows the predictive performance of using clusterings

based directly on demographic variables in terms of respondents gender (2 clusters), country (29 clusters), region (4 clusters), age (86 clusters), age-group (3 clusters), and region and age-group (12 clusters, as in Figure 3).

The predictive log-likelihood of standard LCA reaches a maximum at around $K = 20$ clusters and drops when K is increased. The predictive performance of Bayesian LCA is higher than maximum likelihood LCA. It increases until around $K = 30$ clusters and remains stable from then on, demonstrating that Bayesian LCA is less prone to over-fitting.

A key advantage of the predictive approach is that it can be used for model order selection. While the Bayesian LCA does not over-fit for larger K, the predictive log-likelihood levels off which indicates that a model of order $K = 20$ is complex enough to capture most of structure within data. In order to interpret and compare inferred clusterings, a less complex clustering might be desired. Figure 6 shows that already at $K = 5$ the LCA models infers clusterings that better describe data than segmentations obtained by heuristic based methods or demographics - even for much higher K.

All the evaluated demographic-based clusterings clearly provide better than random predictions, except for *gender* which is only slightly better than random; however, as might be expected, the predictive performance obtained using only demographics is inferior to the clustering approaches, that are fitted on training data to optimally capture the statistical structure in the data.

When creating a clustering such that the training data is partitioned according to nationality, i.e 29 clusters (which is in the vicinity of the optimal number of clusters identified by the models), the predictive log-likelihood becomes significantly lower than for the two models. The same is true for the other demographics markers, clearly indicating that the models identify information beyond demographics and that there is statistical support for this in the data.

Though both K-means and Hierarchical clustering performs significantly better than using the demographic markers they perform worse than LCA, especially for low K.

The predictive performance for the Bayesian model reaches a maximum at a higher number of clusters than for standard LCA. This indicates that the Bayesian framework allows the model to reveal a more complex structure by partitioning the data into more clusters. From the maximum, the predictive performance of the Bayesian model remains constant for higher number of clusters, while it drops for the non-Bayesian LCA, allowing it to fit the data into more clusters while not over-fitting to the training data. However figure 5 shows that a higher number of clusters results in more small clusters, while the distribution of the larger clusters seems to remain rather constant.

Bayesian LCA allows for empty clusters and hence do not guarantee that the data will be split into all K clusters. This allows the model to be less sensitive to the selected number of clusters. If K is high enough, the Bayesian model will simply not partition the data into all the available clusters and the predictive performance will remain high.

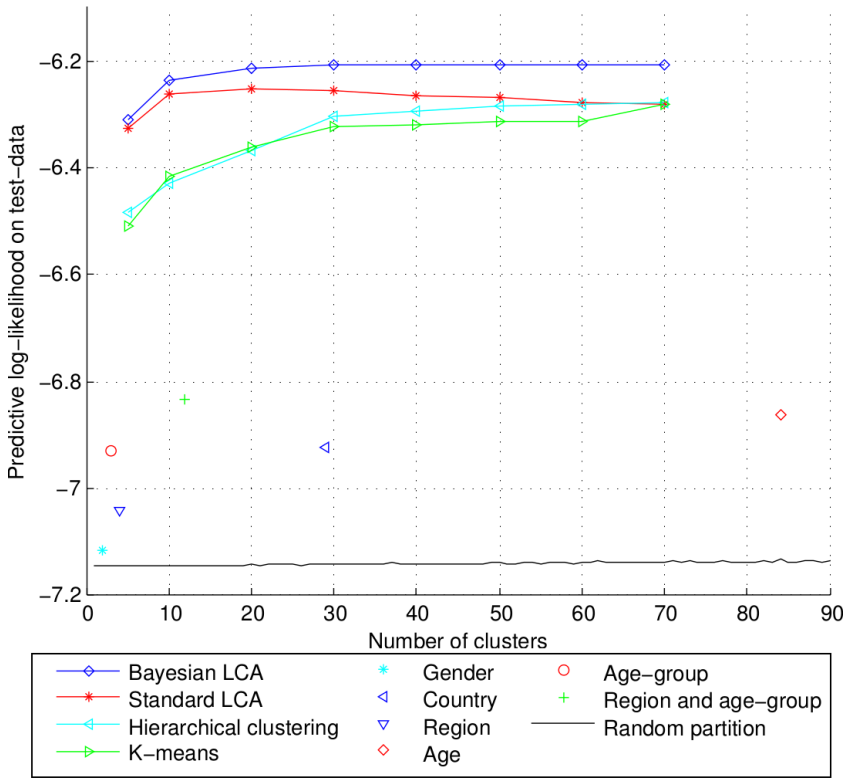


Figure 6. The average predictive log-likelihood versus the number of clusters. For the LCA models, the results are averaged over 5 random restarts of the inference procedures.

Response patterns of LCA clusterings

Table 2 shows proportions of positive responses to the PVQ21 question items indicated by members of the respective clusters for $K = 5$. The figure demonstrates that the Bayesian model has extracted clusters that are highly similar to those extracted by standard LCA. For both models, the largest cluster is dominated by positive responses to all 21 questions, while the other clusters are dominated by negative responses to the questions associated with one or two subordinate values. Cluster 2 is negative towards 'openness to change' and 'self-enhancement'. Cluster 3 is negative towards 'self-enhancement' and slightly negative towards 'conservation'. Cluster 4 is slightly negative to all subordinate values except 'openness to change'. Cluster 5 is negative towards 'openness to change' and 'self-enhancement'. For $K = 5$, all clusters have positive responses to the questions associated to the subordinate value 'self-transcendence'.

Table 3 shows proportions of positive responses for clustering with $K = 20$. Here the models can recover clusters associated to negative responses for the 'self-transcendence' subordinate value. For Bayesian LCA this is seen in cluster 14 to 20 and for standard LCA this is in cluster 13 and 16 to 20. As seen from figure 5 these clusters are fairly small, in

	Bayesian LCA					Standard LCA				
Question \ Cluster	1	2	3	4	5	1	2	3	4	5
Openness to change										
Creative	98%	84%	94%	92%	56%	98%	85%	93%	92%	53%
New things	98%	67%	90%	88%	30%	98%	66%	91%	88%	28%
Good time	99%	66%	88%	90%	35%	98%	67%	88%	90%	31%
Own decisions	100%	95%	96%	94%	77%	100%	95%	96%	94%	75%
Adventures	85%	13%	54%	77%	7%	87%	7%	57%	79%	6%
Fun	98%	55%	90%	89%	30%	98%	55%	92%	89%	25%
Conservation										
Secure	99%	99%	83%	77%	88%	99%	99%	82%	74%	88%
Follow rules	93%	92%	62%	48%	72%	93%	92%	59%	45%	73%
Modest	94%	95%	91%	57%	88%	94%	95%	92%	53%	87%
Safety	99%	99%	87%	77%	85%	99%	99%	85%	76%	84%
Behavior	99%	98%	85%	63%	85%	99%	99%	84%	57%	84%
Traditions	96%	96%	79%	65%	82%	96%	96%	79%	61%	82%
Self-transcendence										
Equality	99%	98%	98%	88%	90%	99%	98%	97%	87%	89%
Understand	99%	97%	99%	83%	86%	99%	97%	99%	82%	86%
Help people	100%	99%	100%	90%	89%	100%	99%	100%	89%	88%
Friends	100%	100%	100%	94%	92%	100%	100%	100%	93%	92%
Nature	100%	99%	99%	88%	91%	100%	99%	99%	86%	90%
Self-enhancement										
Rich	83%	44%	14%	69%	14%	85%	40%	10%	70%	13%
Abilities	98%	82%	50%	89%	27%	99%	83%	43%	90%	23%
Success	99%	81%	52%	88%	23%	99%	82%	44%	90%	18%
Respect	95%	84%	49%	75%	47%	95%	85%	45%	76%	47%

Table 2. Proportions of positive responses to th PVQ21 question items, when modelling with $K = 5$. The clusters are ordered according to size.

total consisting of 3.6% and 3.8% of all respondents respectively for the two models. In the training data just 2.14% of the respondents answers positive to only one or two of the questions associated with 'self-transcendence'. This indicates that the added complexity of $K = 20$ allows the models to contain small clusters and capture such specific response patterns. The added complexity also allows the models to identify clusters that do not contain negative responses for entire subordinate value groups, but also simply for a single or a few questions across value groups.

For both $K = 5$ and $K = 20$ a relatively large proportions of sample belongs to the first cluster which do not indicate specific value priorities.

Subordinate value positions of LCA clusterings

Based on averaging the responses for the four subordinate value groups, the clusters can be positioned on the two value dimensions of the Schwartz circle: *conservation* to *openness to change* and *self-enhancement* to *self-transcendence*.

This is illustrated in Figure 7 for Bayesian LCA with $K = 5$. This figure is comparable to Figure 3 in Magun et al. (2015) where five clusters named as "Growth", "Strong Personal Focus", "Weak Personal Focus", "Strong Social Focus" and "Weak Social Focus" are plotted. Though the first, largest cluster in Figure 7 do not indicate any

Table 3. Proportions of positive responses to th PVQ21 question items, when modelling for $K = 20$. The clusters are ordered according to size.

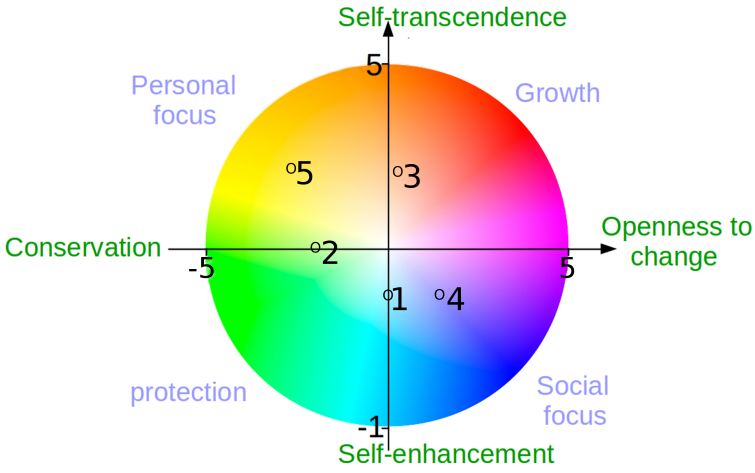


Figure 7. The clusters inferred by Bayesian LCA for $K = 5$ clusters, projected onto the two dimensions spanned by the value axis of the Schwartz circle.

specific value, the figure identifies clusters for ”social focus”, ”strong personal focus”, ”weak personal focus” and ”growth”. Similar to Magun et al. (2015) the figure shows no clustering for the ”protection” value and forms the value diagonal: ”social focus” to ”personal focus”.

The overview of the clusters positioned on the value dimensions for various K for both models are plotted in Figure 8. The clusters are ordered according to their size, with cluster 1 containing most respondents. The clusters are positioned according to their average score with a coloured area spans the standard deviation for the cluster. The figure shows that the clusters are scattered on the three focus areas: Growth, Personal focus and Social focus similar to Magun et al. (2015). For $K = 5$ the inferred clusters are very similar for the two models. For higher K the position of the cluster centroids differs, yet the distribution of clusters seem to span similar areas of the value space and form the diagonal ”Personal focus” to ”Social focus”.

Demographics of LCA clusterings

Figure 9 depicts how the populations in the respective 29 countries are distributed across the clusters identified by the Bayesian LCA. The countries are separated into four regions ”Nordic countries”, ”West European countries”, ”Mediterranean countries” and ”Post-Communist countries” as classified in Magun et al. (2015). The figure shows that the countries are internally diverse, as the individual countries populations are split across the clusters and tend to be represented in all clusters. Especially for $K = 5$ it is evident that there are regional differences. Nordic and West European countries are more represented in the growth and social focus cluster (cluster 3 and 4), while the Post-Communist countries seem to be better represented in the clusters leaning towards

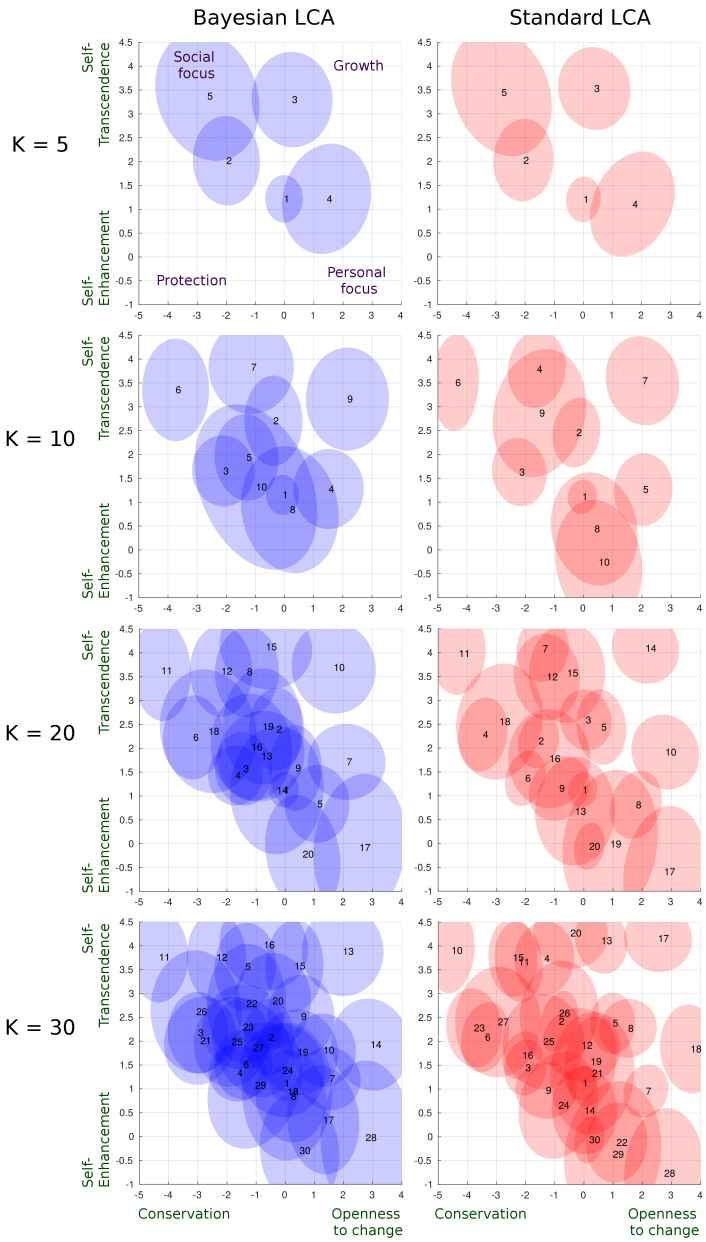


Figure 8. The extracted clusters of the two models for various K, projected on the two value axis of the Schwartz circle shown in figure 1. The clusters are positioned according to their average score and numbered according to their size. The coloured area around a number spans the standard deviation for the cluster.

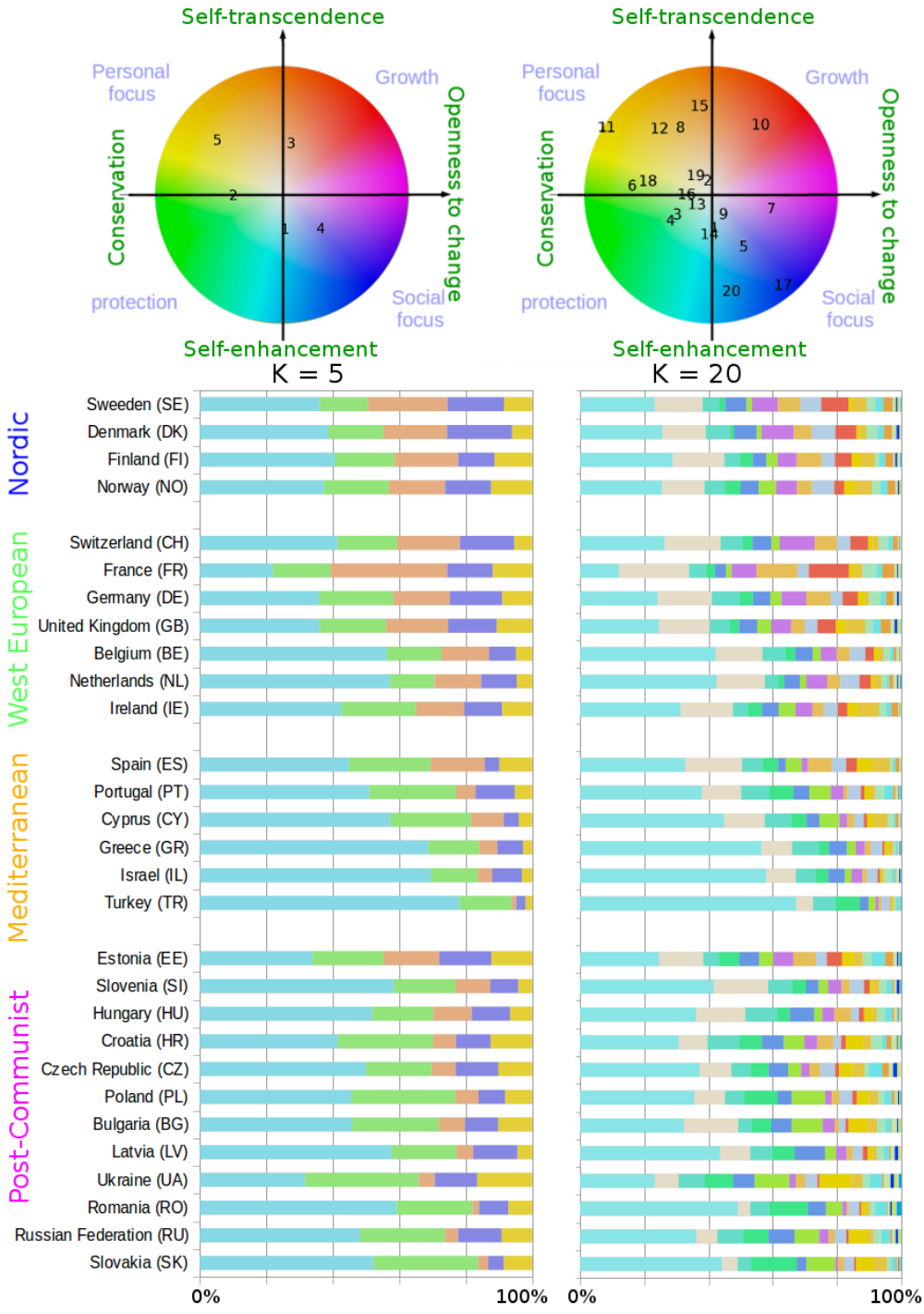


Figure 9. Distribution of the residents of the 29 countries within the inferred clusters. The figure compares Bayesian LCA with $K = 5$ and $K = 20$ and illustrates the clusters projected onto the two value dimensions of the Schwartz circle.

protection and personal focus (cluster 2 and 5). The same effect is evident for $K = 20$ where Mediterranean and Post-Communist countries are less represented in the bigger growth-oriented clusters (cluster 7 and 10).

This is also captured by the inferred clusterings, as shown in Figure 10. The figure illustrates how the age-groups are distributed across the inferred clusters. The age-groups seem to be similarly distributed when considering all countries as well as the four region individually.

For $K = 5$ the percentage of respondents in the individual clusters depends on the age-group, with the same tendency accross regions. For $K = 20$ the percentage of respondents in cluster 2 differs between regions though it seems to be similar for the three agegroups within all regions. This indicates that the cluster represents a grouping of respondents, who share values independent of age. Cluster 2 is positioned slightly towards *self-transcendence* while being neutral on the *openness to change* to *conservation* axis. From Table 3 we identify that cluster 2 leans slightly towards *self-transcendence* mainly as a result of negative responses to the importance of rich question.

The inferred clusters of both Bayesian and traditional LCA are represented in all value dimensions, indicating that the models are capable of capturing both the between-country and within-country diversity simultaneously.

Conclusion

In this paper we have addressed the problem of comparing how well different segmentations capture the underlying structure in data, by comparing models using the presented predictive framework. This highlights the key advantage of using a generative probabilistic approach, namely that the model provides a statistically salient evaluation of model fit based on evaluating the predictive log-likelihood on hold out data.

Prediction remains the intuitive and natural data-driven approach for measuring and evaluating performance. In this paper we have demonstrated how the predictive framework benefits sociological studies, by allowing comparisons of the model fit of different segmentation methods for identifying group-structure within human values survey data from the forth round of European Social Study (ESS-4). In particular we compared the predictive performance of segmentations based on demographics and Latent Class Analysis (LCA). Though demographics can characterize some of the structure within data, LCA showed a significant better predictive performance, highlighting that groups within the data are not only based on demographics and can not be identified by demographic markers alone.

Comparing the predictive performance we found that LCA performs better than both hierarchical and K-means clustering. The Bayesian version performs best, and does not seem to overfit the training data for larger number of clusters. The the extracted clusterings of Bayesian and standard LCA are however fairly similar. The inferred clusters of both Bayesian and traditional LCA are represented in all value dimensions. LCA is capable of capturing both between-region and within-region diversity simultaneously. Interpreting the clusters in terms of demographic composition, similar correlations are identified as expected from the demographics alone. However,

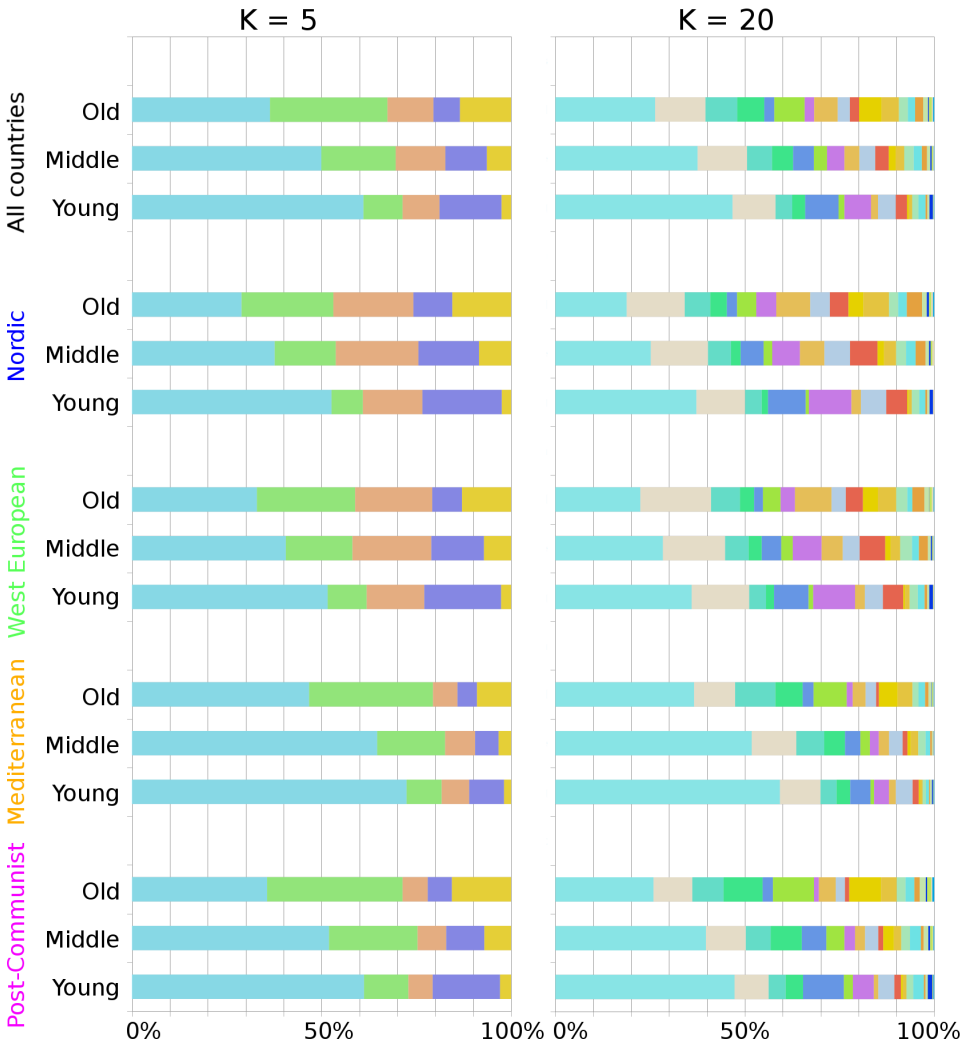


Figure 10. Distribution of age-groups within the inferred clusters. The figure compares Bayesian LCA with $K = 5$ and $K = 20$ and shows the distribution for all countries as well as the four regions individually. The figure is based on the training data, split according to age (young: younger than 35 years, middle: 35 to 54 years old, old: older than 54 years). Respondents with no associated age information in the data are ignored.

the predictive performance of modelling significantly outperforms simply clustering according to demographics. This indicates that the data statistically support to describe the population in greater details than simply determined by demographics, and that mixture modelling to some extend is capable of quantifying this structure.

The introduction of globalized communication technologies has provided means for common value priorities to easily be promoted across geographical boundaries.

Our segmentation results appropriately reflect the trend of value priorities commonly shared across regions while still observing regional and national specific characteristics, as expected when human value priorities transcends geographical and local social boundaries and divides into more complex personality types shared across borders or cultures.

Appendix

Bayesian LCA model specification

The goal of the clustering problem is to split the N respondents into K clusters, based on the structure of their response patterns for the $Q = 21$ questions. Consider the data set represented by a binary matrix as in expression (1). The Bernoulli distribution is a probabilitity distribution of a binary random variable x, that takes on the value 1 with probability θ and 0 with probability $1 - \theta$, resulting in the following probability density function:

$$p(x|\theta) = \theta^x(1-\theta)^{1-x} \qquad \text{for } x \in \{0,1\} \tag{9}$$

The probability that a given respondent i answers positive to question q is set to follow the Bernoulli distribution:

$$x_{iq} \sim Bernoulli(\eta_{\ell q}), \tag{10}$$

In the simplest case, this probability can be considered to be the same and independent for all respondents and only depend on the particular question q and the latent cluster $\ell = z_i$ that the respondent belongs to; parametrised by $\eta_{\ell q}$. For K clusters, we can consider a mixture of K Bernoulli distributions, such that the likelihood of all the responses $\boldsymbol{X}$ becomes:

$$p(\boldsymbol{X}|\boldsymbol{\eta}) = \prod_{i=1}^{N}\prod_{q=1}^{Q} \eta_{z_i,q}{}^{x_{i,q}}(1-\eta_{z_i,q})^{1-x_{i,q}} = \prod_{\ell=1}^{K}\prod_{q=1}^{Q} \eta_{\ell,q}{}^{N^+_{\ell,q}}(1-\eta_{\ell,q})^{N^-_{\ell,q}}, \tag{11}$$

where $N^+_{\ell,q}$ and $N^-_{\ell,q}$ respectively denotes the total sum of positive and negative responses for all respondents in cluster ℓ to question q. Conceptually, a reasonable choice for $\eta_{\ell,q}$ should be based on the ratio of positive and negative responses to the question q for the respondents in cluster ℓ.

The Beta-distribution is given by:

$$p(\mu|\beta^+,\beta^-) = \frac{1}{B(\beta^+,\beta^-)}\mu^{\beta^+-1}(1-\mu)^{\beta^--1} \quad \text{for } \mu \in [0,1] \ , \ \beta^+,\beta^- > 0, \tag{12}$$

where $B()$ denotes the beta function, with $\Gamma()$ being the gamma function:

$$B(\beta^+,\beta^-) \ = \ \int_0^1 \theta^{\beta^+-1}(1-\theta)^{\beta^--1}\,d\theta \ = \ \frac{\Gamma(\beta^+)\Gamma(\beta^-)}{\Gamma(\beta^+ + \beta^-)}$$

The prior belief in the ratio of positive and negative responses is set to follow the Beta-function, which mathematically convenient acts as conjugate prior to the Bernoulli likelihood:

$$\eta_{\ell q} \sim Beta(\beta_q^+, \beta_q^-), \tag{13}$$

where the believed ratio of the response ratio only depends on the particular question q.

The conjugacy of the two distributions means that the posterior distribution of the model belongs to the same family of distributions as the Beta-prior. This conjugacy allows $\boldsymbol{\eta}$ to be analytically marginalized (integrated), revealing the following joint distribution:

$$Bernoulli(\boldsymbol{X}|\boldsymbol{\eta}) \cdot Beta(\boldsymbol{\eta}|\boldsymbol{\beta}^+, \boldsymbol{\beta}^-) \;=\; \prod_{\ell=1}^{K} \prod_{q=1}^{Q} \frac{B(N_{\ell,q}^+ + \beta_q^+, N_{\ell,q}^- + \beta_q^-)}{B(\beta_q^+, \beta_q^-)}, \tag{14}$$

Let $\pi = \{\pi_1, ..., \pi_K\}$ denote the probability distribution for any respondent to belong to the clusters, such that $p(z_i = \ell|\pi) = \pi_\ell$. To allow for flexible cluster sizes, the clustering $\boldsymbol{z}$ of the respondents into K clusters is based on the Dirichlet distribution:

$$p(\pi|c) = \frac{1}{\mathrm{B}(c)} \prod_{k=1}^{K} \pi_k^{c_k - 1}, \quad \text{where} \quad \mathrm{B}(c) = \frac{\Gamma(\sum_{k=1}^{K} c_k)}{\prod_{k=1}^{K} \Gamma(c_k)} \tag{15}$$

With no prior information to pick one cluster above another, a symmetric distribution is preferred. With equal concentration parameters: $\frac{\alpha}{K} = c_1 = ... = c_K$, the following joint prior over $\boldsymbol{z}$ and π is obtained:

$$p(\pi, \boldsymbol{z}|c) \;=\; p(\pi|c) \prod_{i=1}^{N} p(z_i|\pi) \;=\; \frac{1}{\mathrm{B}(c)} \prod_{k=1}^{K} \pi_k^{m_k + c_k - 1}, \tag{16}$$

where m_k is the number of respondents in cluster k. Marginalizing over π reveals the following effective prior over $\boldsymbol{z}$

$$p(\boldsymbol{z}|\alpha) \;=\; \int p(\pi, \boldsymbol{z}|c)\, d\pi \;=\; \frac{\Gamma(\alpha)}{\Gamma(\alpha + N)} \prod_{k=1}^{K} \frac{\Gamma(\frac{\alpha}{K} + m_k)}{\Gamma(\frac{\alpha}{K})}, \tag{17}$$

being the Pólya distribution depending on the single parameter α.

Finally, the joint posterior distribution of the model is obtained when joining (14) and (17):

$$p(\boldsymbol{X}, \boldsymbol{z}|\alpha, \beta^+, \beta^-) \quad = \\ \frac{\Gamma(\alpha)}{\Gamma(\alpha + N)} \prod_{k=1}^{K} \frac{\Gamma(\frac{\alpha}{K} + m_k)}{\Gamma(\frac{\alpha}{K})} \prod_{\ell=1}^{K} \prod_{q=1}^{Q} \frac{B(N_{\ell,q}^+ + \beta_q^+, N_{\ell,q}^- + \beta_q^-)}{B(\beta_q^+, \beta_q^-)} \tag{18}$$

Inference in the Bayesian LCA model

The model parameters are inferred by a sequence of Markov Chain Monte Carlo methods.

The clustering is inferred using a procedure of both full and restricted Gibbs sampling. In the full Gibbs sampling, all respondents are iteratively proposed reassigned to the K clusters based on the posterior distribution of the cluster-assignment for the particular respondent, obtained by Bayes' theorem for equation (18):

$$p(z_i = \ell | \boldsymbol{X}, \boldsymbol{z}^{\setminus i}, \alpha, \beta^+, \beta^-) = \frac{p(\boldsymbol{X}, \boldsymbol{z}^{\setminus i}, z_i = \ell | \alpha, \beta^+, \beta^-)}{\sum_{k=1}^{K} p(\boldsymbol{X}, \boldsymbol{z}^{\setminus i}, z_i = k | \alpha, \beta^+, \beta^-)} \tag{19}$$

In the restricted Gibbs sampling, two clusters are randomly selected and three Gibbs sweep are perform, restricted to re-partitioning the nodes within the selected clusters.

The model contains a number of hyper-parameters:

$$\alpha \ , \ \beta_1^+, ..., \beta_Q^+ \ , \ \beta_1^-, ..., \beta_Q^-$$

They are all sampled independently using a Metropolis-Hastings procedure. Here, proposals for each of the parameters are drawn from a Gaussian distribution centred at the current value of the parameter and with variance 1. The proposals are accepted or rejected according to the Metropolis-Hastings accepting criterion, being the ratio of how likely the model is when using the proposed parameter value compared to the current value.

For all experiments, our sampling strategy consists of 1000 sweeps of the following sampling procedures. First a complete Gibbs sweep over all respondents is performed followed by three proposals of the restricted Gibbs sampling and 10 Metropolis-Hastings proposals for each of the hyperparameters.

References

Allenby GM, Arora N and Ginter JL (1998) On the heterogeneity of demand. *Journal of Marketing Research* 35: 384–389.

Allport GW (1961) Pattern and growth in personality .

Berzofsky ME, Biemer PP and Kalsbeek WD (2014) Local dependence in latent class analysis of rare and sensitive events. *Sociological Methods & Research* 43(1): 137–170.

Bilsky W, Janik M and Schwartz SH (2011) The structural organization of human values: Evidence from three rounds of the european social survey (ess). *Journal of Cross-Cultural Psychology* 42(5): 759–776.

Davidov E, Schmidt P and Billiet J (2011) *Cross-cultural analysis: Methods and applications*. Routledge.

Eid M, Langeheine R and Diener E (2003) Comparing typological structures across cultures by multigroup latent class analysis a primer. *Journal of Cross-Cultural Psychology* 34(2): 195–210.

Feather NT (1995) Values, valences, and choice: The influences of values on the perceived attractiveness and choice of alternatives. *Journal of personality and social psychology* 68(6): 1135–1151.

Finch WH and Bronk KC (2011) Conducting confirmatory latent class analysis using m plus. *Structural Equation Modeling* 18(1): 132–151.

Fraley C and Raftery AE (2002) Model-based clustering, discriminant analysis, and density estimation. *Journal of the American statistical Association* 97(458): 611–631.

Glückstad FK, Schmidt MN and Mørup M (2016) Examination of heterogeneous societies: Identifying subpopulations by contrasting cultures. *Journal of Cross-Cultural Psychology* .

Jowell R, Roberts C, Fitzgerald R and Eva G (2007) *Measuring attitudes cross-nationally: Lessons from the European Social Survey*. Sage.

Kankaraš M, Moors G and Vermunt JK (2010) Testing for measurement invariance with latent class analysis. *Cross-cultural analysis: Methods and applications* : 359–384.

Kluckhohn C (1951) The study of culture. in d. lerner & h.d. lasswell (eds.). *The policy sciences* : 86–101.

Lanza ST, Collins LM, Lemmon DR and Schafer JL (2007) Proc lca: A sas procedure for latent class analysis. *Structural Equation Modeling* 14(4): 671–694.

Magun V and Rudnev M (2008) Basic human values of russians: similarities and dissimilarities with the other european countries. *The Russian Public Opinion Herald* (93): 33–58.

Magun V and Rudnev M (2015) Basic human values of the russians. *Culture Matters in Russiaand Everywhere: Backdrop for the Russia-Ukraine Conflict* : 431–450.

Magun V, Rudnev M and Schmidt P (2015) Within-and between-country value diversity in europe: A typological approach. *European Sociological Review* : 1–14.

McCutcheon A (1987) Latent class analysis. sage university paper series on quantitative applications in the social sciences, no. 07-064 .

Moors G and Vermunt J (2007) Heterogeneity in post-materialist value priorities. evidence from a latent class discrete choice approach. *European Sociological Review* 23(5): 631–648.

Morris C (1956) Varieties of human value .

Muthén B (2008) Latent variable hybrids: Overview of old and new models. *Advances in latent variable mixture models* 1: 1–25.

Nylund KL, Asparouhov T and Muthén BO (2007) Deciding on the number of classes in latent class analysis and growth mixture modeling: A monte carlo simulation study. *Structural equation modeling* 14(4): 535–569.

Picard F (2007) An introduction to mixture models. *Statistics for Systems Biology, Research Report* (7).

Rastelli R and Friel N (2016) Optimal bayesian estimators for latent variable cluster models. *arXiv preprint arXiv:1607.02325* .

Rokeach M (1973) *The nature of human values*. Free press New York.

Rudnev M, Magun V and Schmidt P (2014) The stability of the value typology of europeans: Testing invariance with confirmatory latent class analysis. *Higher School of Economics Research Paper No. WP BRP* .

Schwartz SH (2003) A proposal for measuring value orientations across nations. *Questionnaire Package of the European Social Survey* : 259–290.

Schwartz SH (2007) Value orientations: Measurement, antecedents and consequences across nations. *Measuring attitudes cross-nationally: Lessons from the European Social Survey* : 169–203.

Schwartz SH (2012) An overview of the schwartz theory of basic values. *Online readings in Psychology and Culture* 2(1).

Schwartz SH, Melech G, Lehmann A, Burgess S, Harris M and Owens V (2001) Extending the cross-cultural validity of the theory of basic human values with a different method of measurement. *Journal of cross-cultural psychology* 32(5): 519–542.

Smith PB and Schwartz SH (1997) Values. *Handbook of cross-cultural psychology, 2nd ed., vol.* 3: 77–118.

Szakolczai A and Füstös L (1998) Value systems in axial moments a comparative analysis of 24 european countries. *European Sociological Review* 14(3): 211–229.

Tyler TR and Schuller RA (1991) Aging and attitude change. *Journal of personality and social psychology* 61(5): 689–697.

Veroff J, Reuman D and Feld S (1984) Motives in american men and women across the adult life span. *Developmental psychology* 20(6): 1142–1158.

Ward JH (1963) Hierarchical grouping to optimize an objective function. *Journal of the American statistical association* 58(301): 236–244.

Bibliography

[Albers et al., 2013] Albers, K. J., Moth, A. L. A., Mørup, M., and Schmidt, M. N. (2013). Large scale inference in the infinite relational model: Gibbs sampling is not enough. In *2013 IEEE International Workshop on Machine Learning for Signal Processing (MLSP)*, pages 1–6. IEEE.

[Aldous, 1985] Aldous, D. J. (1985). Exchangeability and related topics. In *École d'Été de Probabilités de Saint-Flour XIII—1983*, pages 1–198. Springer.

[Amunts et al., 2016] Amunts, K., Ebell, C., Muller, J., Telefont, M., Knoll, A., and Lippert, T. (2016). The human brain project: Creating a european research infrastructure to decode the human brain. *Neuron*, 92(3):574–581.

[Andersen et al., 2012] Andersen, K. W., Mørup, M., Siebner, H., Madsen, K. H., and Hansen, L. K. (2012). Identifying modular relations in complex brain networks. In *Machine Learning for Signal Processing (MLSP), 2012 IEEE International Workshop on*, pages 1–6. IEEE.

[Angelino et al., 2014] Angelino, E., Kohler, E., Waterland, A., Seltzer, M., and Adams, R. P. (2014). Accelerating mcmc via parallel predictive prefetching. *arXiv preprint arXiv:1403.7265*.

[Baars and Gage, 2010] Baars, B. J. and Gage, N. M. (2010). *Cognition, brain, and consciousness: Introduction to cognitive neuroscience*. Academic Press.

[Beal, 2003] Beal, M. J. (2003). *Variational algorithms for approximate Bayesian inference*. University of London London.

[Behrens, 2008] Behrens, G. R. (2008). *Mode jumping in MCMC*. PhD thesis, University of Bath.

[Berger, 1997] Berger, J. O. (1997). Some recent developments in bayesian analysis, with astronomical illustrations. In *Statistical Challenges in Modern Astronomy II*, pages 15–48. Springer.

[Berger, 2000] Berger, J. O. (2000). Bayesian analysis: A look at today and thoughts of tomorrow. *Journal of the American Statistical Association*, 95(452):1269–1276.

[Bessiere et al., 2013] Bessiere, P., Mazer, E., Ahuactzin, J. M., and Mekhnacha, K. (2013). *Bayesian programming*. CRC Press.

[Brooks, 2003] Brooks, S. P. (2003). Bayesian computation: a statistical revolution. *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, 361(1813):2681–2697.

[Buchanan et al., 2012] Buchanan, C. R., Gorgolewski, K., Pernet, C. R., Storkey, A. J., and Bastin, M. E. (2012). Quantifying the intra-and inter-subject variability of whole-brain structural networks from diffusion mri. In *International Society for Magnetic Resonance in Medicine 20th Annual Meeting & Exhibition*.

[Budge et al., 1992] Budge, K. G., Peery, J. S., and Robinson, A. C. (1992). High-performance scientific computing using c++. Technical report, Sandia National Labs., Albuquerque, NM (United States).

[Bullmore and Sporns, 2009] Bullmore, E. and Sporns, O. (2009). Complex brain networks: graph theoretical analysis of structural and functional systems. *Nature Reviews Neuroscience*, 10(3):186–198.

[Byrd et al., 2008] Byrd, J. M., Jarvis, S. A., and Bhalerao, A. H. (2008). Reducing the run-time of mcmc programs by multithreading on smp architectures. In *Parallel and Distributed Processing, 2008. IPDPS 2008. IEEE International Symposium on*, pages 1–8. IEEE.

[Byrd et al., 2010] Byrd, J. M., Jarvis, S. A., and Bhalerao, A. H. (2010). On the parallelisation of mcmc by speculative chain execution. In *Parallel & Distributed Processing, Workshops and Phd Forum (IPDPSW), 2010 IEEE International Symposium on*, pages 1–8. IEEE.

[Cary et al., 1997] Cary, J. R., Shasharina, S. G., Cummings, J. C., Reynders, J. V., and Hinker, P. J. (1997). Comparison of c++ and fortran 90 for object-oriented scientific programming. *Computer Physics Communications*, 105(1):20–36.

[Chambers, 2000] Chambers, J. M. (2000). Users, programmers, and statistical software. *Journal of Computational and Graphical Statistics*, 9(3):404–422.

[Clauset et al., 2008] Clauset, A., Moore, C., and Newman, M. E. (2008). Hierarchical structure and the prediction of missing links in networks. *Nature*, 453(7191):98–101.

[Cooper, 1990] Cooper, G. F. (1990). The computational complexity of probabilistic inference using bayesian belief networks. *Artificial intelligence*, 42(2-3):393–405.

[Cormen et al., 2001] Cormen, T. H., Leiserson, C. E., Rivest, R. L., and Stein, C. (2001). *Introduction to Algorithms*. Massachusetts Institute of Technology Press and McGraw-Hill.

[Cowles and Carlin, 1996] Cowles, M. K. and Carlin, B. P. (1996). Markov chain monte carlo convergence diagnostics: A comparative review. *Journal of the American Statistical Association*, 91(434):883–904.

[Daducci et al., 2012] Daducci, A., Gerhard, S., Griffa, A., Lemkaddem, A., Cammoun, L., Gigandet, X., Meuli, R., Hagmann, P., and Thiran, J.-P. (2012). The connectome mapper: an open-source processing pipeline to map connectomes with mri. *PLoS ONE*, 7(12).

[Dagum and Menon, 1998] Dagum, L. and Menon, R. (1998). Openmp: an industry standard api for shared-memory programming. *IEEE computational science and engineering*, 5(1):46–55.

[Dayan and Abbott, 2005] Dayan, P. and Abbott, L. (2005). *Theoretical Neuroscience: Computational And Mathematical Modeling of Neural Systems*. Massachusetts Institute of Technology Press.

[Demidov et al., 2013] Demidov, D., Ahnert, K., Rupp, K., and Gottschling, P. (2013). Programming cuda and opencl: A case study using modern c++ libraries. *SIAM Journal on Scientific Computing*, 35(5):C453–C472.

[Desikan et al., 2006] Desikan, R. S., Ségonne, F., Fischl, B., Quinn, B. T., Dickerson, B. C., Blacker, D., Buckner, R. L., Dale, A. M., Maguire, R. P., Hyman, B. T., et al. (2006). An automated labeling system for subdividing the human cerebral cortex on mri scans into gyral based regions of interest. *Neuroimage*, 31(3):968–980.

[Destrieux et al., 2010] Destrieux, C., Fischl, B., Dale, A., and Halgren, E. (2010). Automatic parcellation of human cortical gyri and sulci using standard anatomical nomenclature. *Neuroimage*, 53(1):1–15.

[Dos Reis and Stroustrup, 2011] Dos Reis, G. and Stroustrup, B. (2011). A principled, complete, and efficient representation of c++. *Mathematics in Computer Science*, 5(3):335–356.

[Duane et al., 1987] Duane, S., Kennedy, A. D., Pendleton, B. J., and Roweth, D. (1987). Hybrid monte carlo. *Physics letters B*, 195(2):216–222.

[Feinberg et al., 2010] Feinberg, D. A., Moeller, S., Smith, S. M., Auerbach, E., Ramanna, S., Glasser, M. F., Miller, K. L., Ugurbil, K., and Yacoub, E. (2010). Multiplexed echo planar imaging for sub-second whole brain fmri and fast diffusion imaging. *PloS one*, 5(12):e15710.

[Fischl et al., 2004] Fischl, B., Van Der Kouwe, A., Destrieux, C., Halgren, E., Ségonne, F., Salat, D. H., Busa, E., Seidman, L. J., Goldstein, J., Kennedy, D., et al. (2004). Automatically parcellating the human cerebral cortex. *Cerebral cortex*, 14(1):11–22.

[Fortunato and Barthelemy, 2007] Fortunato, S. and Barthelemy, M. (2007). Resolution limit in community detection. *Proceedings of the National Academy of Sciences*, 104(1):36–41.

[Gelman et al., 2014] Gelman, A., Carlin, J. B., Stern, H. S., Dunson, D. B., Vehtari, A., Rubin, D. B., and Rubin, D. B. (2014). *Bayesian data analysis*. Taylor & Francis, 3 edition.

[Gelman and Rubin, 1992a] Gelman, A. and Rubin, D. B. (1992a). Inference from iterative simulation using multiple sequences. *Statistical science*, 7(4):457–511.

[Gelman and Rubin, 1992b] Gelman, A. and Rubin, D. B. (1992b). A single series from the gibbs sampler provides a false sense of security. *Bayesian statistics*, 4:625–631.

[Gerlach and Kneis, 2003] Gerlach, J. and Kneis, J. (2003). Generic programming for scientific computing in c++, javatm, and c. In *International Workshop on Advanced Parallel Processing Technologies*, pages 301–310. Springer.

[Gershman and Blei, 2012] Gershman, S. J. and Blei, D. M. (2012). A tutorial on bayesian nonparametric models. *Journal of Mathematical Psychology*, 56(1):1–12.

[Ghahramani, 2013] Ghahramani, Z. (2013). Bayesian non-parametrics and the probabilistic approach to modelling. *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, 371(1984):20110553.

[Ghosh and Deriche, 2016] Ghosh, A. and Deriche, R. (2016). A survey of current trends in diffusion mri for structural brain connectivity. *Journal of neural engineering*, 13(1):011001.

[Gillispie et al., 2000] Gillispie, C. C., Fox, R., and Grattan-Guinness, I. (2000). *Pierre-Simon Laplace, 1749-1827: a life in exact science*. Princeton University Press.

[Glasser et al., 2016] Glasser, M., Coalson, T., Robinson, E., Hacker, C., Harwell, J., Yacoub, E., Ugurbil, K., Anderson, J., Beckmann, C., Jenkinson, M., Smith, S., and Essen, D. (2016). A multi-modal parcellation of human cerebral cortex. *Nature*, 536:171–178.

[Griffin and Holmes, 2010] Griffin, J. and Holmes, C. (2010). Computational issues arising in bayesian nonparametric hierarchical models. *Bayesian Nonparametrics*, pages 208–222.

[Hagmann, 2005] Hagmann, P. (2005). *From diffusion MRI to brain connectomics*. PhD thesis, Ecole Polytechnique Fédérale de Lausanne.

[Hagmann et al., 2008] Hagmann, P., Cammoun, L., Gigandet, X., Meuli, R., Honey, C. J., Wedeen, V. J., and Sporns, O. (2008). Mapping the structural core of human cerebral cortex. *PLoS Biology*, 6(7):1479–1493.

[Hansen et al., 2011] Hansen, T. J., Mørup, M., and Hansen, L. K. (2011). Nonparametric co-clustering of large scale sparse bipartite networks on the gpu. In *Machine Learning for Signal Processing (MLSP), 2011 IEEE International Workshop on*, pages 1–6. IEEE.

[Herlau et al., 2012] Herlau, T., Mørup, M., Schmidt, M. N., and Hansen, L. K. (2012). Modelling dense relational data. In *Machine Learning for Signal Processing (MLSP), 2012 IEEE International Workshop on*, pages 1–6. IEEE.

[Jain and Neal, 2004] Jain, S. and Neal, R. M. (2004). A split-merge markov chain monte carlo procedure for the dirichlet process mixture model. *Journal of Computational and Graphical Statistics*, 13(1):158–182.

[Jorgenson et al., 2015] Jorgenson, L. A., Newsome, W. T., Anderson, D. J., Bargmann, C. I., Brown, E. N., Deisseroth, K., Donoghue, J. P., Hudson, K. L., Ling, G. S., MacLeish, P. R., et al. (2015). The brain initiative: developing technology to catalyse neuroscience discovery. *Phil. Trans. R. Soc. B*, 370(1668):20140164.

[Kadry, 2014] Kadry, S. (2014). History of the modern probability philosophy. *Open Journal of Philosophy*, 2014.

[Kemp et al., 2006] Kemp, C., Tenenbaum, J. B., Griffiths, T. L., Yamada, T., and Ueda, N. (2006). Learning systems of concepts with an infinite relational model. In *AAAI*, volume 3, pages 381–388.

[Lazar, 2013] Lazar, N. (2013). The big picture: Big data computing. *CHANCE*, 26(2):28–32.

[Liben-Nowell and Kleinberg, 2007] Liben-Nowell, D. and Kleinberg, J. (2007). The link-prediction problem for social networks. *Journal of the American society for information science and technology*, 58(7):1019–1031.

[Lunn et al., 2009] Lunn, D., Spiegelhalter, D., Thomas, A., and Best, N. (2009). The bugs project: Evolution, critique and future directions. *Statistics in medicine*, 28(25):3049–3067.

[Lunn et al., 2000] Lunn, D. J., Thomas, A., Best, N., and Spiegelhalter, D. (2000). Winbugs - a bayesian modelling framework: concepts, structure, and extensibility. *Statistics and computing*, 10(4):325–337.

[Mahani and Sharabiani, 2015] Mahani, A. S. and Sharabiani, M. T. (2015). Simd parallel mcmc sampling with applications for big-data bayesian analytics. *Computational Statistics & Data Analysis*, 88:75–99.

[Mengersen et al., 1999] Mengersen, K. L., Robert, C. P., and Guihenneuc-Jouyaux, C. (1999). Mcmc convergence diagnostics: a reviewww. *Bayesian statistics*, 6:415–440.

[Miller et al., 2009] Miller, K., Griffiths, T. L., and Jordan, M. I. (2009). Nonparametric latent feature models for link prediction. In *Advances in neural information processing systems*, pages 1276–1284.

[Moeller et al., 2010] Moeller, S., Yacoub, E., Olman, C. A., Auerbach, E., Strupp, J., Harel, N., and Uğurbil, K. (2010). Multiband multislice ge-epi at 7 tesla, with 16-fold acceleration using partial parallel imaging with application to high spatial and temporal whole-brain fmri. *Magnetic Resonance in Medicine*, 63(5):1144–1153.

[Mørup et al., 2014] Mørup, M., Glückstad, F. K., Herlau, T., and Schmidt, M. N. (2014). Nonparametric statistical structuring of knowledge systems using binary feature matches. In *Machine Learning for Signal Processing (MLSP), 2014 IEEE International Workshop on*, pages 1–6. IEEE.

[Mørup et al., 2010] Mørup, M., Madsen, K., Dogonowski, A.-m., Siebner, H., and Hansen, L. K. (2010). Infinite relational modeling of functional connectivity in resting state fmri. In *Advances in neural information processing systems*, pages 1750–1758.

[Mørup and Schmidt, 2012] Mørup, M. and Schmidt, M. N. (2012). Bayesian community detection. *Neural computation*, 24(9):2434–2456.

[Neal, 2011] Neal, R. M. (2011). Mcmc using hamiltonian dynamics. *Handbook of Markov Chain Monte Carlo*, pages 113–162.

[Newman, 2003] Newman, M. E. (2003). The structure and function of complex networks. *SIAM review*, 45(2):167–256.

[Nowicki and Snijders, 2001] Nowicki, K. and Snijders, T. A. B. (2001). Estimation and prediction for stochastic blockstructures. *Journal of the American Statistical Association*, 96(455):1077–1087.

[Ogawa et al., 1990] Ogawa, S., Lee, T.-M., Kay, A. R., and Tank, D. W. (1990). Brain magnetic resonance imaging with contrast dependent on blood oxygenation. *Proceedings of the National Academy of Sciences*, 87(24):9868–9872.

[Oldford, 1988] Oldford, R. W. (1988). Object-oriented software representations for statistical data. *Journal of econometrics*, 38:227–246.

[Oldford, 1990] Oldford, R. W. (1990). Software abstraction of elements of statistical strategy. *Annals of Mathematics and Artificial Intelligence*, 2(1):291–307.

[Palla et al., 2012] Palla, K., Knowles, D., and Ghahramani, Z. (2012). An infinite latent attribute model for network data. *Proceedings of the 29th International Conference on Machine Learning*, pages 1607–1614.

[Patil et al., 2010] Patil, A., Huard, D., and Fonnesbeck, C. J. (2010). Pymc: Bayesian stochastic modelling in python. *Journal of statistical software*, 35(4):1.

[Reislev et al., 2012] Reislev, N. L., Ptito, M., Kupers, R., Siebner, H. R., and Dyrby, T. B. (2012). Alterations of the inferior longitudinal fasciculus in congenital and late blindness. In *ISMRM 2012 meeting abstract*, volume 3706, pages 10950–10960.

[Richiardi et al., 2013] Richiardi, J., Achard, S., Bunke, H., and Van De Ville, D. (2013). Machine learning with brain graphs: predictive modeling approaches for functional imaging in systems neuroscience. *IEEE Signal Processing Magazine*, 30(3):58–70.

[Sawitzki, 1996] Sawitzki, G. (1996). Extensible statistical software: On a voyage to oberon. *Journal of Computational and Graphical Statistics*, 5(3):263–283.

[Schmidt and Mørup, 2013] Schmidt, M. N. and Mørup, M. (2013). Nonparametric bayesian modeling of complex networks: An introduction. *IEEE Signal Processing Magazine*, 30(3):110–128.

[Schoett, 1986] Schoett, O. (1986). Data abstraction and the correctness of modular programming.

[Setsompop et al., 2012] Setsompop, K., Gagoski, B. A., Polimeni, J. R., Witzel, T., Wedeen, V. J., and Wald, L. L. (2012). Blipped-controlled aliasing in parallel imaging for simultaneous multislice echo planar imaging with reduced g-factor penalty. *Magnetic Resonance in Medicine*, 67(5):1210–1224.

[Snijders and Nowicki, 1997] Snijders, T. A. and Nowicki, K. (1997). Estimation and prediction for stochastic blockmodels for graphs with latent block structure. *Journal of classification*, 14(1):75–100.

[Sporns, 2009] Sporns, O. (2009). The human connectome: linking structure and function in the human brain. In *Johansen-Berg, H and Behrens TEJ (eds.), Diffusion MRI: From Quantitative Measurement to in vivo Neuroanatomy*, pages 309–332. Academic Press, Amsterdam.

[Sporns et al., 2005] Sporns, O., Tononi, G., and Kötter, R. (2005). The human connectome: a structural description of the human brain. *PLoS Computational Biology*, 1(4):e42.

[Stan Development Team, 2012] Stan Development Team, . (2012). Stan: A c++ library for probability and sampling. *http://mc-stan.org*.

[Strid, 2010] Strid, I. (2010). Efficient parallelisation of metropolis–hastings algorithms using a prefetching approach. *Computational Statistics & Data Analysis*, 54(11):2814–2835.

[Strohmaier et al., 2005] Strohmaier, E., Dongarra, J. J., Meuer, H. W., and Simon, H. D. (2005). Recent trends in the marketplace of high performance computing. *Parallel Computing*, 31(3):261–273.

[Stroustrup, 2013] Stroustrup, B. (2013). *The C++ Programming Language*. Addison-Wesley.

[Van Essen et al., 2013] Van Essen, D. C., Smith, S. M., Barch, D. M., Behrens, T. E., Yacoub, E., Ugurbil, K., Consortium, W.-M. H., et al. (2013). The wu-minn human connectome project: an overview. *Neuroimage*, 80:62–79.

[Von Neumann, 1945] Von Neumann, J. (1945). First draft of a report on the edvac. Technical report, Department of Computer Science, Michigan State University, prepared for U.S. Army Ordinance Department under Contract W-670-ORD-4926.

[Xu et al., 2012] Xu, J., Moeller, S., Strupp, J., Auerbach, E., Chen, L., Feinberg, D., Ugurbil, K., and Yacoub, E. (2012). Highly accelerated whole brain imaging using aligned-blipped-controlled-aliasing multiband epi. In *Proceedings of the 20th Annual Meeting of ISMRM*, volume 2306.

[Xu et al., 2006] Xu, Z., Tresp, V., Yu, K., and Kriegel, H.-P. (2006). Learning infinite hidden relational models. *Uncertainity in Artificial Intelligence (UAI2006)*.

[Yuste and Church, 2014] Yuste, R. and Church, G. M. (2014). The new century of the brain. *Scientific American*, 310(3):38–45.

[Zhu et al., 2009] Zhu, S., Yu, K., and Gong, Y. (2009). Stochastic relational models for large-scale dyadic data using mcmc. In *Advances in Neural Information Processing Systems*, pages 1993–2000.